PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
H04N 1/41

A1

(11) International Publication Number: **WO 97/21302**

(43) International Publication Date: 12 June 1997 (12.06.97)

(21) International Application Number: PCT/US96/19388

(22) International Filing Date: 5 December 1996 (05.12.96)

(30) Priority Data:

60/008,294	8 December 1995 (08.12.95)	US
60/023,569	7 August 1996 (07.08.96)	US
60/024,804	29 August 1996 (29.08.96)	US

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, US, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

(71) Applicant *(for all designated States except US)*: TRUSTEES OF DARTMOUTH COLLEGE [US/US]; 11 Rope Ferry Road, Hanover, NH 03755 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: DANSKIN, John, M. [US/US]; 12 Fletcher Circle, Hanover, NH 03755 (US). DAVIS, Geoffrey [US/US]; Apartment 2W, 26 E. Wheelock, Hanover, NH 03755 (US).

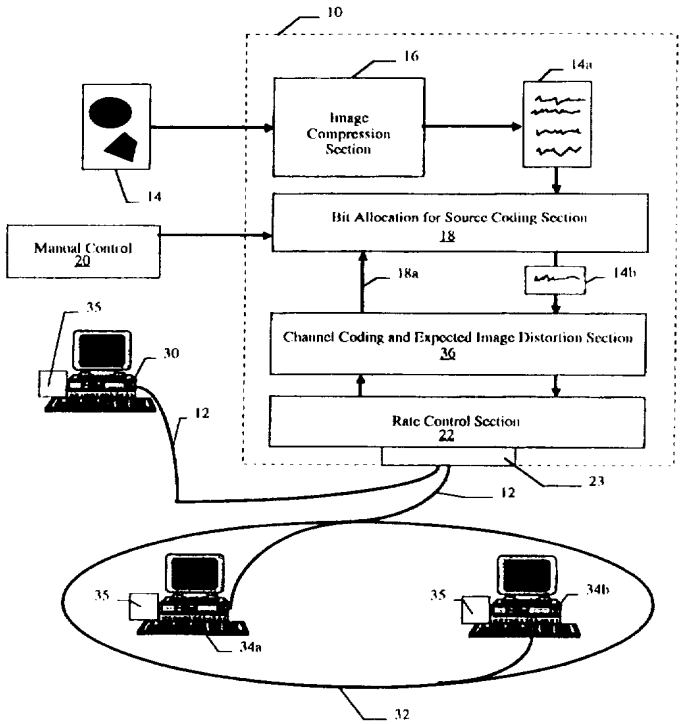
(74) Agents: VOCK, Curtis, A. et al.; Lappin & Kusmer L.L.P., Two Hundred State Street, Boston, MA 02109 (US).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: FAST LOSSY INTERNET IMAGE TRANSMISSION APPARATUS AND METHODS

(57) Abstract

Fast lossy Internet image transmission ("FLIIT") systems and methods are provided for transmitting images (14), such as world wide web graphics, over the Internet. Forward error correction, added to an image during compression, enables the subsequent reconstruction of fragments lost during transmission by purposefully concentrating image bits within the portions of the image that have the greatest overall visual impact. Image fragments that are lost during transmission have little noticeable effect, and no time is spent on retransmitting lost fragments, such as in TCP/IP. FLIIT eliminates retransmission delays by strategically shielding important parts of subband coded images through forward error correction. Each subband is decomposed into a series of bitplanes ordered from the most significant to the least significant. An optimization procedure determines the subset of bitplanes to transmit as well as the number of bits to spend on forward error correction for each bitplane, recognizing that different bits in compressed images such as JPEG have different contributions to image fidelity. FLIIT also assesses current network conditions and adjusts transmission rates so as to accommodate network traffic: keeping the total transmission bits constant, more bits are adjusted to data during low network congestion, while more bits are adjusted to redundancy during high network congestion. A decoding section within a receiver unit, e.g., personal computer, decodes the transmitted image upon arrival across the Internet, providing a 2-4 factor improvement in speed over existing image transfers with the same quality.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	**GB**	United Kingdom	**MW**	Malawi
AT	Austria	**GE**	Georgia	**MX**	Mexico
AU	Australia	**GN**	Guinea	**NE**	Niger
BB	Barbados	**GR**	Greece	**NL**	Netherlands
BE	Belgium	**HU**	Hungary	**NO**	Norway
BF	Burkina Faso	**IE**	Ireland	**NZ**	New Zealand
BG	Bulgaria	**IT**	Italy	**PL**	Poland
BJ	Benin	**JP**	Japan	**PT**	Portugal
BR	Brazil	**KE**	Kenya	**RO**	Romania
BY	Belarus	**KG**	Kyrgystan	**RU**	Russian Federation
CA	Canada	**KP**	Democratic People's Republic of Korea	**SD**	Sudan
CF	Central African Republic			**SE**	Sweden
CG	Congo	**KR**	Republic of Korea	**SG**	Singapore
CH	Switzerland	**KZ**	Kazakhstan	**SI**	Slovenia
CI	Côte d'Ivoire	**LI**	Liechtenstein	**SK**	Slovakia
CM	Cameroon	**LK**	Sri Lanka	**SN**	Senegal
CN	China	**LR**	Liberia	**SZ**	Swaziland
CS	Czechoslovakia	**LT**	Lithuania	**TD**	Chad
CZ	Czech Republic	**LU**	Luxembourg	**TG**	Togo
DE	Germany	**LV**	Latvia	**TJ**	Tajikistan
DK	Denmark	**MC**	Monaco	**TT**	Trinidad and Tobago
EE	Estonia	**MD**	Republic of Moldova	**UA**	Ukraine
ES	Spain	**MG**	Madagascar	**UG**	Uganda
FI	Finland	**ML**	Mali	**US**	United States of America
FR	France	**MN**	Mongolia	**UZ**	Uzbekistan
GA	Gabon	**MR**	Mauritania	**VN**	Viet Nam

Fast Lossy Internet Image Transmission Apparatus and Methods

Related Applications

This is a continuing application of Provisional Application No. 60/008,294, filed on December 8, 1995, and entitled "Fast Lossy Internet Image Transmission Apparatus and Methods," of Provisional Application No. 60/023,569, entitled "Fast Lossy Internet Image Transmission Apparatus and Methods" and filed on August 6, 1996, and of Provisional Application No. 60/024804, entitled "Fast Lossy Internet Image Transmission Apparatus and Methods" and filed on August 29, 1996, each of which is hereby incorporated by reference.

Background

World Wide Web requests are the single largest consumer of Internet bandwidth, comprising roughly 25% of all bytes sent. *See,* Georgia Tech Graphics, Visualization, & Usability Center, "Third degree polynomial curve fitting for bytes transferred per month by service," NSFNET Backbone Statistics Page, August 1995, http://www.cc.gatech.edu/gvu/stats/NSF/-merit.html. Images, most of which are examined for a only a few seconds, undoubtedly constitute the bulk of the ten terabytes of current monthly Web requests. For such interactive applications as web browsers, the responsiveness gained from rapid image transmission is more important than perfect image fidelity, since many images are already distorted by lossy compression, and since relatively few images are closely examined.

The usual method for transmitting images over the Internet is to first compress the images using a lossy scheme such as JPEG, and then to transmit

the compressed images across the intrinsically lossy Internet using the lossless TCP/IP protocol. JPEG and related lossy schemes are very sensitive to bit errors and hence require lossless transmission. The price paid for lossless transmission over a lossy medium, however, is excessively lengthy transmission times due to retransmissions of lost packets.

Specifically, TCP/IP retransmits missing pieces until the image is complete, resulting in inefficiencies and considerable transmission delays. This is particularly true with the growing popularity of the Internet, which has led to increased network congestion and "traffic jams" that cause fragments of images to be lost in transit. Because lossless TCP/IP depends upon retransmission to correct network losses, the transmission time for even relatively short messages can be substantial, particularly during times of heavy network traffic.

Lossless transmission schemes are even more problematic for Internet video broadcasting. Retransmission is impractical with such broadcasting because the receivers will not, in general, experience the same losses. Accordingly, a broadcaster attempting to respond to all of the different losses can be overwhelmed with requests to retransmit lost packets.

A number of strategies have been explored for incorporating redundancy into network packets. In Turner et al., Image transfer: an end-to-end design, SigComm 92, 258-268, for example, a scheme is presented in which errors are corrected by making use of naturally occurring redundancy within images. Image pixels are reordered for transmission in such a way that packet losses cause the loss of isolated pixels rather than of large contiguous blocks of pixels. Missing pixels are reconstructed by applying a filter to neighboring pixels that survive transmission, thereby hiding a limited number of missing

packets when there is high correlation between neighboring pixels. However, when such a correlation does not exist, the technique does not readily mask missing packets.

In a related scheme, a network video transmission scheme proposed by Karlsson et al., Subband coding of video for packet networks, Optical Engineering, 27(7), 574-586 (1988), also makes use of naturally occurring image redundancy for error correction. As above, using intrinsic image redundancy to correct losses remains problematic, since the number of losses that can be sustained is highly image dependent. Furthermore, when efficient compression schemes are used, very little usable redundancy remains for error correction. That is, common image compression techniques typically operate to remove redundant image pixels so that reconstruction of adjacent pixels is ineffective.

In other techniques, the control of transmission errors is obtained by adding redundancy bits to the bitstream rather than by relying solely on naturally occurring redundancy. In Biersack, Performance evaluation of forward error correction in ATM networks, Proceedings of the SIGCOMM 92 Symposium, Baltimore, 248-257 (1992), for example, a technique is presented that evaluates the effect of redundancy addition at a fixed rate to video transmissions over ATM networks. This fixed rate addition of redundancy, however, is inherently inefficient and can obtain mixed results. By way of example, testing of this technique has shown that in heterogeneous traffic scenarios, the loss rates were reduced by several orders of magnitude; but for more homogeneous traffic scenarios, the performance was unchanged or worsened. Further, in homogeneous traffic scenarios, the increase in the network load from the transmission of redundancy bits can cause an increase in the loss rate not compensated for by the error correction.

Another prior art method of adding redundancy is through joint source and channel coding. C. Shannon, in A Mathematical Theory of Communication, Bell System Technical Journal, Vol. 27, pp. 379-423, 623-656 (1948), describes a source-channel separation theorem which states that separate source and channel coding procedures can be made to be just as effective as a joint procedure. Nevertheless, the results of joint source and channel coding are asymptotic and require infinite length messages.

A Lagrange multiplier-based joint source-channel coding scheme for continuous bitstreams has also been developed in the prior art. See, e.g., N. Tanabe at al., Subband image coding using entropy-coded quantization over noisy channels, IEEE Journal on Selected Areas in Communications, 10:5, 926-943 (1992). In this scheme, however, error calculations for continuous streams are extremely complex, and the algorithms presented rely on computationally expensive simulations during bit allocation.

A related source-channel coding scheme for networks, entitled "Priority Encoding Transmission" (hereinafter "PET"), has also been developed in the prior art. See A. Albanese et a., Priority encoding transmission, Proc. 35th Annual Symposium on Foundations of Computer Sciences, Santa Fe, NM, pp. 604-612 (1994); C. Leicher, Hierarchical encoding of MPEG sequences using priority encoding transmission (PET), TR-94-058, ICSI, Berkeley, CA (1994). The implementation of PET for MPEG allows the user to set different levels of error protection for different portions of the MPEG stream, but provides little or no methodology for allocating these levels.

The layered transmission schemes in M. Garrett, Joint source/channel coding of statistically multiplexed real-time services on packet networks, IEEE

Transactions on Networking, 1:1, 71-80 (1993), and E. Posnak et al., Techniques for resilient transmission of JPEG video streams, also make use of prior art joint source-channel coding methods. These layered schemes require networks that treat packets differently according to their priorities. Visually important data is thus sent with a high priority, i.e., with a smaller loss rate, and less important data is sent with a low priority so as to be discarded first by switches during congestion. Accordingly, these schemes require networks capable of providing prioritized handling of packets, a capability that is not always available on the Internet.

Another prior art lossless Internet flow control technique is described in L. Brakmo, TCP Vegas: New techniques for congestion detection and avoidance, Proceedings of the SIGCOMM '94 Symposium, (1994). This TCP Vegas technique achieves rate control by managing the number of packets stored in the network, rather than by forcing losses as TCP Reno does. However, this technique is also problematic in that it has a relatively slow start-up time.

It is desirable to speed up Internet image transmissions so that online users can view, download and operate on images more quickly. This is true regardless of the bandwidth of the online connection, e.g., an Internet connection including a personal computer, modem and phone line operating at 28.8kbps (kilobytes-per-second), a digital telephone line, and/or the ethernet. It is especially desirable to increase the speed of Internet image transmission without significant loss in image quality.

It is, accordingly, an object of the invention to provide apparatus and methods for increasing the speed of Internet image transmissions.

Still another object of the invention is to provide systems and methods for adjusting the transmission speed of Internet images with selectable image quality.

Yet another object of the invention is to provide a fast, lossy Internet image transmission methodology which reduces the problems associated with prior art Internet image transmission methods.

Another object of the invention is to provide an error correction system which speeds up Internet image transmissions in a manner compatible with existing networks and without significant loss of image quality.

These and other objects will become apparent in the description which follows.

Summary of the Invention

The invention provides an efficient method for transmitting images, such as world wide web graphics, over the Internet. In a preferred aspect, the invention makes use of forward error correction ("FEC"), which allows the recipient of an image on the Internet to reconstruct fragments lost during transmission. Preferably, the FEC methodology of the invention is added to an image during compression, and purposefully concentrates image bits within the portions of the image that have the greatest overall visual impact. Accordingly, image fragments that are lost during transmission have little noticeable effect, and no time is spent on retransmitting lost fragments, such as in TCP/IP.

More particularly, in one aspect, the invention provides a fast lossy Internet image transmission (hereinafter "FLIIT") methodology that eliminates retransmission delays by strategically shielding important parts of subband coded images through FEC. Each subband is decomposed into a series of bitplanes ordered from the most significant to the least significant. An optimization procedure, described in more detail below, determines the subset of bitplanes to transmit as well as the number of bits to spend on FEC for each bitplane. Bits are allocated in order to maximize the expected quality of the received image subject to an overall bit budget. The FLIIT methodology recognizes that different bits in compressed images such as JPEG have different contributions to image fidelity. For example, flipping high order bits in the DC channel of a JPEG-compressed image results in a large discernible difference in the decompressed image, whereas flipping low order bits in a high frequency channel has little visual effect. Typically, applying equal amounts of redundancy to protect bits in these two categories is not efficient.

The FLIIT methodology preferably utilizes a first order Markov model of the bursty Internet packet loss structure. The use of the Markov model enables the determination of the effects of network burst errors within parity groups.

In another aspect, the invention incorporates error correction into a standard wavelet-based subband coder. Specifically, the FLIIT methodology of the invention allocates bits between the tasks of encoding image subbands and protecting coded data with FEC. Bits devoted to subband coding correspond to the image to be transmitted, and bits devoted to FEC increase the likelihood of that image arriving intact. This allocation reduces the distortions in the received image, both from compression and network losses, and subject to a constraint on the total bytes transmitted. Accordingly, the FEC bits are concentrated in subbands where losses would be visually catastrophic, while less important subbands receive less protection.

In still other aspects, the invention addresses network issues such as rate, congestion control, and startup. FLIIT methodology allocates a fixed number of bits between redundancy and data depending on the expected loss rate. When the loss rate is high, bits are shifted from data to redundancy, but the total number of transmitted bits remains constant. In the prior art, on the other hand, TCP retransmits more and more packets during heavy congestion because packet loss rates are high, presenting a positive feedback that worsens congestion. In accord with the invention, FLIIT methodology reduces this positive feedback by sending packets with a fixed total number of bits exactly once, trading quantizer resolution for FEC as a function of current network conditions.

Unlike the prior art, one aspect of a system constructed according to the invention includes a server that remembers the last sending rate for each recent connection, eliminating slow startups for repeat connections. Prior art TCP, on the other hand, has a slow startup procedure can take seconds to ramp up to full speed, which can be quite ineffective, particularly with respect compressed images like JPEGs. The invention achieves flow control by managing the number of packets stored in the network, and usually avoids the slow startup associated with TCP by remembering transmission rates for recent connections so that multiple short connections to a single client will only have to pay for one startup.

Another aspect of the invention includes a data determination section that evaluates and decides when to stop waiting for data packets that may have been lost or delayed. The data determination section monitors and assesses the waiting period for packets, and balances that wait period between an insufficient time that risks losing image data, and an excessive time that results in reduced system responsiveness.

In another aspect, the invention includes a burst-loss control module within a Bit Allocation for Source Coding Section which interleaves packets in order to decorrelate burst losses, thereby advantageously utilizing the structure of burst losses to achieve improved transmission fidelity.

Further, in another aspect, the invention can include a subband coder module within a Channel Coding and Expected Image Distortion Section using nested quantization to reduce FEC requirements.

In still another aspect, the invention includes a flow control section which operates to choose an optimal time to stop waiting for lost packets.

Typically, the ideal stopping point is the expected time of arrival of the last packet plus the standard deviation of the interpacket arrival time.

In still another aspect, the invention provides a system which efficiently transmits image data via some redundant transmission such as FEC, which makes serious transmission errors unlikely. The system applies this redundancy selectively, however, since the addition of redundancy increases the amount of information that must be transmitted. Indeed, experiments in FEC redundancy as applied uniformly to ATM video packets have shown a decrease in performance in some cases since the increased network load due to the redundancy can lead to an increase in the packet loss rate. See Biersack, Performance evaluation of forward error correction in ATM networks, Proceedings of the SIGCOMM 92 Symposium, Baltimore, 248-257 (1992). Accordingly, the invention provides certain operational controls, described in more detail below, which function to optimize FEC redundancy in view of network congestion, desired image quality, and/or transmission speed.

The invention thus provides several important advantages over the prior art. First, the invention speeds up Internet image transmissions by a factor of 2 to 4 over TCP while maintaining images of similar overall quality. The invention is also suitable for fast transmission of video over the Internet, and, more importantly, seamlessly coexists with existing TCP connections.

The invention further provides a methodology for obtaining an optimized partitioning of bits between source coding and channel coding for a given set of (1) image subband quantizers, (2) FEC protection levels, and (3) packet loss model. In one aspect, for example, a rate control section is provided to accommodate other lossy Internet media protocols, such as real time voice transmission.

The following articles and book chapters provide useful background to the invention and are, accordingly, incorporated herein by reference: A. Albanese et al., Priority encoding transmission, Proc. 35th Annual Symposium on Foundations of Computer Sciences, Santa Fe, NM, pp. 604-612 (1994); T.C. Bell et al., Text Compression, Prentice Hall, Englewood Cliffs, NJ (1990); E.W. Biersack, Performance evaluation of forward error correction in ATM networks, Proceedings of the SIGCOMM 92 Symposium, Baltimore, 248-257, (1992); L. S. Brakmo et al., TCP Vegas: New techniques for congestion detection and avoidance, Proceedings of the SIGCOMM '94 Symposium (1994); T.M. Cover et al., Elements of Information Theory, John Wiley & Sons, Inc., New York (1991); J. M. Danskin, G. Davis, and X. Song, "Fast Lossy Internet Image Transmission," ACM MultiMedia 95, pp. 321-332 (1995); B. Fox, Discrete optimization via marginal analysis, Management Science 7 13:3, pp. 210-216 (1966); G. Karlsson et al., Subband coding of video for packet networks, Optical Engineering, 27(7), 574-586 (1988); C. Leicher, Hierarchical encoding of MPEG sequences using priority encoding transmission (PET), TR-94-058, ICSI, Berkeley, CA (1994); W. E. Leland et al., On the self-similar nature of ethernet traffic, Proc. SIGCOMM, 183-193, San Francisco (1993); A. S. Lewis et al., Image compression using the 2-D wavelet transform, IEEE Transactions on Image Processing, Vol. **1,** No. **2,** pp. 244-250 (1992); C. E. Shannon, A Mathematical Theory of Communication, Bell System Technical Journal, Vol. 27, pp. 379-423, 623-656 (1948); J. Shapiro, Embedded Image Coding Using Zerotrees of Wavelet Coefficients, IEEE Transactions on Signal Processing, Vol. 41, No. 12, pp. 3445-3462; Y. Shoham and A. Gersho, Efficient bit allocation for an arbitrary set of quantizers, IEEE Trans. Acoustics, Speech, and Sig. Proc., 36:9, 1445-14537 (1988); N. Tanabe, Subband image coding using entropy-coded quantization over noisy channels, IEEE Journal on Selected Areas in Communications, 10:5, 926-943, (1992); A. Tanenbaum, Computer Networks, Prentice-Hall, Englewood

Cliffs, N. J. (1981); Turner, Charles J., and Larry L. Peterson, "Image transfer: and end-to-end design," SigComm 92, 258-268; D. Taubman, Multirate 3-D subband coding of video, IEEE Trans. Image Proc., 3(5) (1994); J.D. Villasenor et al., Wavelet filter evaluatio for image compression, IEEE Trans. Image Processing (1995); C.L. Williamson et al., Loss-load curves: Support for rate-based congestion control in high-speed datagram networks, Proceedings of SIGCOMM 91, pp. 17-287 (1991); I. Witten et al., Arithmetic coding for data compression, Communications of the ACM, 30:6, 520-540 (1987).

The invention is next described further in connection with preferred embodiments, and it will become apparent that various additions, subtractions, and modifications can be made by those skilled in the art without departing from the scope of the invention.

Brief Description of the Drawings

A more complete understanding of the invention may be obtained by reference to the drawings, in which:

Figure 1 shows a schematic layout of a system constructed according to the invention;

Figure 1A shows a schematic layout of another system constructed according to the invention, including software modules to enable FLIIT methodology;

Figure 1B illustrates blocks of data sorted by typical redundancy, in accord with the invention;

Figure 1C illustrates message transmission through typical routers on the Internet;

Figure 1D illustrates a graphical distribution of bitplanes and parities for a representative image in accord with the invention;

Figure 2 illustrates a flowchart for encoding images in accord with the invention;

Figure 2A illustrates a flowchart for decoding images in accord with the invention;

Figure 2B illustrates an alternative flowchart for encoding images in accord with the invention;

Figure 2C graphically shows Internet packet drop rate for packets sent between Dartmouth College and Stanford University as a function of time of day;

Figure 2D graphically shows observed and fitted cumulative density functions for packet delays modeled according to the invention;

Figure 3 graphically illustrates the expected and measured PSNR performances of FLIIT methodology, according to the invention, and three fixed parity schemes;

Figures 4A-4C shows experimental results of Lena images transmitted over a transcontinental Internet connection utilizing flat parity schemes with differing image quality reconstruction percentiles;

Figures 5A-5C shows experimental results of Lena images transmitted over a transcontinental Internet connection utilizing FLIIT methodology, with differing image quality reconstruction percentiles, according to the invention;

Figure 5D illustrates the Lena image of Figures 4 and 5 transmitted via TCP/IP;

Figure 6 schematically illustrates one test configuration used to test the system of the invention;

Figure 7 graphically shows the time advantages of transmitting images via FLIIT as compared to TCP for selected image qualities; and

Figures 8A and 8B graphically show the time impact of FLIIT vs. TCP protocols for various network configurations.

Detailed Description of Illustrated Embodiments

Figure 1 illustrates a system 10 constructed according to the invention for transmitting images through the Internet 12. An uncompressed electronic image 14 is first reduced in size by an Image Compression Section 16 so as to produce, for example, lossy JPEG representations 14a of the image 14. The Bit Allocation for Source Coding Section 18 thereafter partitions and transforms the image 14a into a set of subbands ranging from high frequency, fine scales to low frequency, coarse scales so as to minimize image distortions relative to a total allowed number of transmission bits. These transmission bits form the electronic image file 14b with finely quantized coefficients that contribute heavily to image fidelity, e.g., low frequency image components, and coarsely

quantizing coefficients that contribute little to image fidelity, e.g., high frequency edges.

File 14b is thus suitable for transmission through the Internet 12 and to a client receiving unit 30, e.g., a personal computer, and/or through the Internet 12 and into a network 32 that includes a plurality of client receiving units 34a, 34b. Each of the units 30, 34a, 34b has a decoding subsection 35 housed within associated memory, e.g., firmware or application-specific software within random access memory ("RAM"). As described in more detail below, the decoding subsection 35 operates to "reverse" the encoding process provided by sections 16, 18, except that no bit allocation decisions are made and certain image packets are unavailable due to transmission losses along the Internet and network 12, 32, respectively.

System 10 preferably includes a Channel Coding and Expected Image Distortion Section 36, described below, which dynamically allocates bits between source and channel codes depending upon conditions within the network 32. System 10 connects to the Internet 12 through any of the standard interfaces, e.g., an ethernet connection 23.

System 10 can be implemented in several ways. Generally, however, system 10 includes a central processing unit ("CPU") and one or more connected memories, such as shown in Figure 1A. In Figure 1A, system 10' is a computer or server that includes an image compression section 16', a Bit Allocation for Source Coding Section 18', a Channel Coding and Expected Distortion Section 36', and a Rate Control Section 22, each of which represents a software module in active memory 21' within the computer 10'. System 10' connects to the Internet 12' through any one of the known prior art connections, e.g., an ethernet connection 23', and transmits images and

receives packet information from any of the connected users 30' to adjust image transmission characteristics, as described herein. The CPU 27 controls the system 10', including the input and output of image files into internal memory 21'.

Image Compression Section

Image compression such as performed by the image compression section 16, Figure 1, or section 16', Figure 1A, can occur by one of several methods. By way of example, sections 16, 16' can utilize a wavelet transform coding scheme to compress images for transmission along the Internet 12. Although those skilled in the art will appreciate that other compression schemes can be used, the wavelet-based coder is chosen and described herein because of its simplicity and excellent performance at low bit rates. Experimental results yield, without error correction overhead, peak signal-to-noise ratios (PSNR's) to within less than one dB of an embedded zerotree wavelet coder.

With further reference to Figure 1, a discrete wavelet transform is performed by the image compression section 16 on the image 14 by quantizing the coefficients using uniform quantizers, and by coding the resulting coefficients for entropy using an arithmetic coder. The resolution of the quantizers is determined by a Lagrange multiplier procedure or other optimization procedure describe in more detail below. One suitable transform is a 9/7-tap biorthogonal filter set used in experiments and as described in J.D. Villasenor et al., IEEE Trans. Image Processing (1995).

Bit Allocation for Source Coding Section

The discrete wavelet transform performed by the Image Compression Section 16 results in a compressed image 14a. The Bit Allocation for Source Coding Section 18 thereafter partitions the image 14a into a set of subbands ranging from fine scales, i.e., high frequency, to coarse scales, i.e., low frequency. In natural images, the bulk of the visually important information is concentrated in the coarse-scale subbands, with the fine scale subbands contributing primarily to sharp edge effects. In accord with the invention, the Bit Allocation for Source Coding Section 18 transforms the image 14a into image representation 14b by finely quantizing coefficients that contribute heavily to image fidelity and coarsely quantizing others. Determining the quantization resolution of each subband is a key feature of the Bit Allocation for Source Coding Section 18. More particularly, section 18 performs a tradeoff between quantization error and total storage cost, and allocates quantizer resolutions to obtain minimal distortion for a given bit expenditure. The total bit expenditure can be set by two principal ways: through manual control 20, e.g., a computer and keyboard connected for communication with the system 10, or through feedback determinations of the Rate Control Section 22, each of which is described in more detail below.

The Bit Allocation for Source Coding Section 18 first selects one of a family of quantizers Q_0 ... Q_K for each image subband. The quantizers are arranged from coarsest (Q_0) to finest (Q_K) and have bin widths scaled according to the range R_j of coefficients in each subband. By way of example, certain of the experiments described below employ quantizers Q_K with $\{2^k - 1\}_{0\le k\le 10}$ uniformly spaced bins. Quantizer bins are distributed symmetrically about 0, since wavelet coefficients are known *a priori* to be symmetrically distributed about the origin, and the bins for quantizer Q_K when quantizing subband j

have width $2R_j / (2^k - 1)$, where R_j is the maximum magnitude of a coefficient in subband j. Quantized values are preferably decoded to the center of each quantizer bin.

In an alternative quanitization, section 18 can utilize a family of quantizers such as described in D. Taubman et al., Multirate 3-D subband coding of video, IEEE Trans. Image Proc., 3(5) (1994), whereby a family of nested Q_K quantizers, $2^k - 1$ bins, are used: one bin of width $(2^{-k+1}) R_j$ is centered at the origin; and the other $2^k - 2$ bins are spaced uniformly and symmetrically around the center bin, each with width $2^{-k} R_j$. This family of quantizers has the important property that quantizer bins are nested, i.e. each bin of Q_K can be decomposed into either two or three bins in Q_{K+1}. The output of the quantizer Q_K can be expressed as a string of refinements $(r_0, r_1, \ldots, r_k)$, where each of the r_i's is a 0, 1, or 2. The sets of refinements are essentially the bitplanes of the coefficients ordered from the most significant bit to the least significant bit. This family of nested quantizers permits fine control of the distribution of redundancy so as to vary the protection at the bitplane rather than the coefficient level.

The Bit Allocation for Source Coding Section 18 also determines image distortion during the allocation of bits to the subbands. By way of example, a mean squared error function can be used to assess distortion. This choice also permits comparison with other algorithms; however, the mean squared error function functions equally well with perceptually weighted metrics such known to those in the art. See, e.g., S. Lewis et al., Image compression using the 2-D wavelet transform, IEEE Transactions on Image Processing, Vol. 1, No. 2, pp. 244-250 (1992).

In particular, for the mean squared error function, let $D_j(k)$ be the total squared error incurred in quantizing the wavelet coefficients in subband j with quantizer Q_K, and let $C_j(k)$ be the cost in bits of representing the corresponding entropy-coded quantized values. For an image decomposed into n subbands, the Bit Allocation for Source Coding Section 18 computes to identify a vector $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ of quantizer indices so that the total distortion $D_{total}(\mathbf{q}) = \sum_{j=0}^{n} D_j(q_j)$ is minimized subject to the constraint that the total cost in bits, $C_{total}(\mathbf{q}) = \sum_{j=0}^{n} C_j(q_j)$ is less than or equal to some given bit budget C_{max}. Section 18 thus seeks a minimization over $\mathbf{q} \in Q$ where Q is a given set of valid vectors of quantizer indices.

Marginal analysis, as known to those skilled in the art, see, e.g., B. Fox, Discrete optimization via marginal analysis, Management Science 7 13:3, pp. 210-216 (1966), provides one algorithm suitable for solving this minimization problem. In particular, the Bit Allocation for Source Coding Section 18 initializes the vector of quantizer resolutions **q** to the coarsest configuration, (0, 0, ..., 0), and sets the number of remaining bits to allocate to C_{max}. Allocation then proceeds iteratively as follows: for each subband, the cost and distortion changes resulting from refining the subband's quantizer by one increment is computed. All the subbands for which quantizer refinement is possible are considered, providing that the cost of refinement does not exceed the total remaining bits to allocate. If there are no such subbands, the Bit Allocation for Source Coding Section 18 terminates the algorithm. Otherwise, it finds subband j for which quantizer refinement yields the largest reduction in distortion per bit, increments the corresponding q_j, and subtracts the cost of the refinement from the total remaining bits.

Marginal analysis in accord with the invention thus yields an optimal bit allocation when cost and distortion functions are convex. Marginal analysis is also very fast relative to the cost of the transform, requiring at most nK iterations to converge, where n is the number of subbands and K is the number of quantizers in the family.

Those skilled in the art should appreciate that other bit allocation techniques can be used in accord with the invention. For example, the minimization problem solved by Section 18 over $\mathbf{q} \in Q$ can be solved by Lagrangian techniques as opposed to the marginal analysis described above. In Shoham et al., Efficient bit allocation for an arbitrary set of quantizers, *IEEE Trans. Acoustics, Speech, and Sig. Proc.*, 36:9, 1445-1453 (1988), an algorithm is described which solves the minimization problem. Specifically, the algorithm teaches that an unconstrained minimum of $C_{total}(\mathbf{q}) + \lambda D_{total}(\mathbf{q})$ is also the solution to a constrained problem of the form required. The unconstrained problems are easier to solve; but the value of λ must be determined for the appropriate constrained problem. The constrained problem is thus transformed into a search through a family of unconstrained problems; and the algorithm of Shoham et al. gives appropriate bit allocations for the minimization problem to be solved by Section 18.

Channel Coding and Expected Image Distortion Section

One objective of system 10 is to reduce or minimize the image distortion incurred in quantizing transform coefficients. Transmission of an image 14b over a network introduces a second source of distortion: network packet losses. In accord with the invention, the Channel Coding and Expected Image Distortion Section 36 controls quantization error by adaptively allocating quantizer resolution within the Bit Allocation for Source Coding Section 18

via communication line 18a. In the same way, the Channel Coding and Expected Image Distortion Section 36 controls packet loss errors by selectively adding redundancy to the bitstream transmitted on the Internet 12. The image 14a has already incurred loss during a lossy compression technique, e.g., such as through JPEG, and can generally withstand some additional loss during transmission, provided that those lost bits are not visually important. Because system 10 performs both source and channel coding jointly, the Channel Coding and Expected Image Distortion Section 36 knows the relative values of the bits within the image 14b, and thereby provides an extension of the above-described bit allocations by incorporating expected transmission losses into the distortion function and the costs of redundancy into the cost function. Specifically, the Channel Coding and Expected Image Distortion Section 36 finds an optimized partition of bits into source and channel codes.

The distortion variance can be controlled by adjusting the packet loss model used by the bit allocation algorithm. For example, by numerically increasing the assumed loss probability p_{loss}, the distortion variance can be controlled by adjusting the loss probability assumed in the optimizer. For example, numerically increasing the loss probability beyond the network's true packet loss rate has the effect of shifting bits from data to redundancy, which in turn increases the quantization distortion at a given bit-rate and also increases the protection against lost packets. Since the distortion variance functionally depends upon lost packets, increasing the degree of redundancy reduces the variance and increases image consistency.

The problem thus addressed by the Channel Coding and Expected Image Distortion Section 36 is that of transmitting images as a collection of packets of bits of a maximum size S over the Internet 12 and network 32. The Channel Coding and Expected Image Distortion Section 36 has two separate properties

for classes of network protocols: first, packets can be delivered out of order, so that each packet contains a unique identifier; and secondly, the contents of all packets are verified during transmission. Packets are generally lost for one of two reasons: a node somewhere on the Internet 12 and/or network 32 runs out of buffer space and drops the packet, or the packet is corrupted and fails a verification procedure somewhere in transit. Because of section 36's first property, i.e., that each packet contains a unique identifier, system 10 knows exactly which packets have been lost. Because of section 36's second property, i.e., that the contents of packets are verified during transmission, system 10 assumes that all packets which are delivered are error-free because they have passed the protocol's verification procedure.

To reduce decoded image variance and to facilitate the packing of subbands into equally sized network packets, e.g., such as 576 bytes each, the Channel Coding and Expected Image Distortion Section 36 breaks subbands (or subband bitplanes) into blocks of smaller memory sizes, each of which is preferably a maximum of 150 bytes. To reduce the visual impact of any losses, the Channel Coding and Expected Image Distortion Section 36 distributes pixels into these blocks through interleaving. All of the bitplanes of a subband are interleaved in the same way. In accord with the invention, blocks from a subband which represent different bitplanes, but which derive from the same image pixels, belong to the same interleaving.

The Channel Coding and Expected Image Distortion Section 36 adds redundancy to the image transmission by adding FEC bits to the data stream. Because system 10 can tell which packets have been lost, a single block of FEC bits can protect a group of any number of blocks of data against single-packet loss. The Channel Coding and Expected Image Distortion Section 36 therefore conducts a tradeoff between protection and cost: greater protection of data is

obtained by decreasing the size of the groups protected by FEC blocks, but increased protection increases the total transmission time because of the additional FEC blocks. The Channel Coding and Expected Image Distortion Section 36 thus performs this tradeoff in such a way so as to minimize the expected distortion of the image for a given total number of image bits, and, preferably, as a function of current network congestion, as described below.

In estimating the expected image distortion, the Bit Allocation for Source Coding Section 18 determines a probability of packet loss. To a first approximation, packet losses are independent Bernoulli trials, with losses occurring with probability P. However, studies of network traffic on a network such as network 32 reveal that network traffic is bursty and that these bursts are present across a wide range of time scales. See, W. E. Leland et al., On the self-similar nature of ethernet traffic, Proc. SIGCOMM, 183-193 San Francisco (1993). While it is true that very long bursts of losses in routers do occur, the rate adaptation that takes place in network protocols such as TCP greatly reduces the length of bursts actually experienced by the user. Accordingly, the Bit Allocation for Source Coding Section 18 incorporates bursts into a packet loss model such as through a first order Markov model. Specifically, the Bit Allocation for Source Coding Section 18 denotes a successful transmission by 0 and a loss by 1, and denotes the transition probabilities by $P_{j,k}$, where j, k $\in$ {0, 1} correspond to the fates of two consecutive packets. The steady state loss rate is

$$P_1 = \frac{P_{0,1}}{P_{0,1} + P_{1,0}}$$

and the steady state success rate is $P_0 = 1 - P_1$. For consistency with the Bernoulli model, $P_1 = P$.

To implement FEC according to one embodiment of the invention, data blocks are grouped into parity groups that are formed in of one of three ways by the Bit Allocation for Source Coding Section 18: (1) a parity group consisting of a single unshielded block; (2) a parity group consisting of multiple data blocks shielded by a single parity block; or (3) a parity block consisting of a single data block with multiple replicas. These three types of parity groups provide gradated levels of protection, ranging from minimal, unshielded blocks, to maximal, replicated blocks. For each subband, the Bit Allocation for Source Coding Section 18 determines a level of quantization refinement $q_{j,}$ and a level of parity protection $p_{j,k}$ for each subband bitplane.

The use of the Markov model by section 18 enables the determination of the effects of burst errors within parity groups. A simplifying assumption is made that losses in different parity groups are independent. This between-group independence assumption is relevant only for parity groups containing blocks from the same subband. Section 18 also minimizes the effects of between-group correlation by interleaving the groups when loading packets with blocks.

Section 36 also evaluates the effects of various levels of protection and quantization on coefficients in subband n. For illustration, let $D_{,}$ be the average distortion resulting from setting a coefficient in subband n to zero. Let D_{j} be the average reduction in coefficient distortion given by bitplane j. When a high-order bitplane is lost, all lower order refinements will also be lost, since section 36 conditions the entropy coding of low order bitplanes on the high order values. Let X_{i} correspond to the event that the block containing bitplane i for the coefficient is successfully transmitted. Section 36 then determines the expected distortion for the coefficient by the following relationships.

Let $d = \left\lceil \frac{M}{m} \right\rceil$ be the distance between successive blocks in an interleaved FEC group. Interleaving the FEC group changes the effective success/loss transition probabilities. We compute new transition probabilities $P_{a,b}^{(d)}$ for the interleaved blocks. Let $\sigma = (\sigma_1, \sigma_2,\ldots,\sigma_n)$ be a binary string of length n. Under our Markov model we have $P(\sigma) = P_{\sigma_1} \prod_{k=1}^{n-1} P_{\sigma_k \sigma_{k+1}}$. The new transition probabilities are given by $P_{a,b}^{(d)} = \sum_{k=0}^{d} \sum_{\sigma \in S_{d,k}^{(a,b)}} P(\sigma)$. Let $S_{n,k}^{(a,b)}$ denote the set of all length n binary strings that begin with a, end with b, and contain exactly k 1's. The probability of the sting $S_{n,k}^{(a,b)}$ can be determined recursively as follows. All probabilities $P(S_{2,k}^{(a,b)})$ are zero except $P^{(d)}(S_{2,0}^{(0,0)}) = P_0 P_{0,0}^{(d)}$, $P^{(d)}(S_{2,1}^{(1,0)}) = P_1 P_{1,0}^{(d)}$, $P^{(d)}(S_{2,1}^{(0,1)}) = P_0 P_{0,1}^{(d)}$, and $P^{(d)}(S_{2,2}^{(1,1)}) = P_1 P_{1,1}^{(d)}$. The probability of longer sequences is determined recursively using the relation $P^{(d)}(S_{n,k}^{(a,b)}) = \sum_{i=0}^{k} \sum_{(x,y) \in Q} P_a P^{(d)}(S_{n,i}^{(a,x)}) P_{x,y} P^{(d)}(S_{n,k-i}^{(y,b)})$. The probability P(X) of an unrecoverable loss of a block replicated m times with spacing d is given by $P(X) = P_1 \left(P_{1,1}^{(d)} \right)^{m-1}$. For a group of m blocks with spacing d that contains a single parity block, a block survives if there is at most one lost packet. The probability that any block is lost unrecoverably is $P(X) = \sum_{a=0,1} \sum_{b=0,1} \sum_{l=2}^{m} \frac{l}{m} P^{(d)}\left(S_{n,l}^{(a,b)}\right)$. In general, for an FEC scheme that can recover from up to L losses, we have $P(X) = \sum_{a=0,1} \sum_{b=0,1} \sum_{l=L+1}^{m} \frac{l}{m} P^{(d)}\left(S_{n,l}^{(a,b)}\right)$.

When loading blocks into network packets, section 36 imposes the restriction that no two blocks from the same parity group may occupy the same

packet so that the loss of one packet corresponds to the loss of only one element of a parity group. An additional restriction is imposed to reduce the variance of reconstructed images: no two blocks from different interleavings of the same subband may occupy the same network packet or the same parity group.

For the Bernoulli model, $P(X) = 1 - P$ for unshielded blocks. When a block is replicated, a block is lost only if all copies are lost. Hence, $P(X) = 1 - P^m$, where m is the total number of copies of the block that are transmitted. For groups in which each m blocks are shielded by a single parity block, $P(X) = 1 - P[1 - (1 - P)^m]$, which is one minus the probability of losing the given block and at least one of the other m blocks in the group.

For the Markov model, $P(X) = 1 - P = P_0$ for unshielded blocks. For replicated and parity-shielded blocks, the order in which blocks are transmitted to compute expected distortions must be determined. Accordingly, section 36 decorrelates losses within each parity group by spacing data as far apart as possible in the transmitted bitstream, i.e., at packet intervals of $\left\lceil \frac{M}{m} \right\rceil$ in an m-block group, where M is the total number of packets to be transmitted as determined from the total bit budget.

With $P_{j,k}$ indicating the manner of parity shielding for bitplane k of subband j, the Channel Coding and Expected Image Distortion Section 36 replaces the cost and distortion functions $C_j(q_j)$ and $D_j(q_j)$ with the functions $C_{j,k}(q_{j'}p_{j,k})$ and $D_{j,k}(q_{j'}p_{j,k})$ that incorporate the cost of the parity packets and the expected distortion incurred in transmission. The new cost function $C_j(q_{j'}p_{j,k})$ will equal the old $C_j(q_j)$ plus the number of bits used for the parity blocks. The new distortion function $D_j(q_{j'}p_{j,k})$ is obtained using the expected distortion and

success probabilities described above. As before, marginal analysis or Lagrange multipliers are preferably utilized for bit allocation, though with more choices for each iteration. That is, the Channel Coding and Expected Image Distortion Section 36 either increases the number of bit planes retained for a particular subband, or increases the parity protection for one particular subband or subband bitplane.

Note that smaller groupings are more protective of the data and are thus less likely to distort the image with any packet losses. Because the invention preferably sorts by redundancy levels, Figure 1B illustrates typical data and redundancy groupings. In (a), three data blocks 100 are accompanied by one parity block 102. As such, only one of the data blocks 100 can be lost without image distortion. In (b), one data block 104 is accompanied with one parity block 106. In (c), one data block 108 is three-way replicated with two parity blocks 110. If the groupings are too small, such as in (b), the data 104 is essentially replicated. Each parity block 102, 106, 110 should be as long as the longest block in its parity group.

Figure 1D illustrates a graphical distribution 115 of bitplanes and parities for one representative image (the Lena image discussed in detail in connection with Figures 4 and 5). In Figure 1D, black rectangles represent 64 byte data blocks and the gray rectangles represent the forward area correction (FEC) bits assigned to that data block. Leftmost blocks contain the highest order coefficient bitplanes; rightmost the lowest. The coarsest scale subbands are on the bottom 117 of the chart 115; while the finest subbands are at the top 119. Note that the concentration of FEC bits is larger for higher order coefficient bitplanes and for coarser scale subbands.

Those skilled in the art should appreciate that the expected distortion can be viewed and analyzed in other ways. For example, consider a subband consisting of n data blocks. Let D_q be the average quantization error incurred per block, let D_m be the error incurred in replacing all coefficients in a data block by the quantized subband mean, and let D_z be the error incurred in replacing all coefficients in a data block by zero. Since $D_q \leq D_m \leq D_z$, the zero-replacement is the worst-case scenario.

Data and parity blocks from each subband can therefore be distributed so that no two blocks from the same band are contained in the same network packet. Hence, losses of data blocks can be modeled as independent events. Every block transmitted, and every block that is lost and recovered produces an average error of D_q. If the subband mean is available, i.e., if at least one packet from the group is successfully transmitted, then the lost blocks produce an average error of D_m; otherwise lost blocks produce an average error of D_z. The expected distortion for a band consisting of n data blocks is thus

$$E(D) = nD_q + np_{unrecoverable}(D_m - D_q) + np^n_{unrecoverable}(D_z - D_m),$$

where $p_{unrecoverable}$ is the probability of an unrecoverable packet loss.

Encoding and Decoding

With reference to Figure 2, the steps for encoding and decoding transmitted images between the system 10 and any of the users 30, 34a, 34b are as follows. The encoding occurs in the system 10, prior to transmission on the Internet, and the decoding takes place after transmission and within any of the decoding sections 35.

More particularly, system 10 encodes images by the following steps. First, in step 40, the Image Compression Section 16 applies a compression algorithm, e.g., a wavelet subband decomposition, to the image 14 to form a compressed image representation 14a. Thereafter, in step 42, the Bit Allocation for Source Coding Section 18 decomposes each subband in image 14a into a series of nested quantizers, i.e., bitplanes, and, in step 44, determines which bitplanes to send, and how much redundancy to assign to each bitplane. The resulting image file from Section 18 is shown as image 14b.

In step 46, the Channel Coding and Expected Image Distortion Section 36 then interleaves the bitplanes into blocks of some fixed size, i.e., the size after compression. For example, blocks of 150 bytes each can be used. Even though smaller blocks could be formed to reduce image variance, by spreading losses more evenly, smaller blocks also introduce overhead. Thus, interleaving visually decorrelates losses, although it has no effect on system 10's quality metric. Interleaving has little or no effect on compression performance because the interleaving occurs after the subband decomposition.

In step 48, the Channel Coding and Expected Image Distortion Section 36 compresses the formed blocks using an arithmetic coder. Nested quantization can be coded in the same total number of bits as a non-nested quantization of the same bins by (i) coding in decreasing bitplane magnitude order, and by (ii) using the high order bits for a transformed pixel as a context selecting frequency tables for low order bits.

Finally, in step 50, the Channel Coding and Expected Image Distortion Section 36 adds redundancy to the image and packs blocks into network packets. Blocks of the same protection level are grouped into parity groups of

appropriate size, such as shown in Figure 1B. For example, 576 bytes per network packet is the current maximum size of unfragmented Internet transmissions. Accordingly, 576-byte network packet sizes are preferably used with the invention; though those skilled in the art should appreciate that other packet sizes can be used, particularly if Internet protocol changes. Sorting by size is useful because the parity block is as large as the largest data block in the group. Blocks in the same FEC group are preferably spaced out as much as possible to take advantage of burst losses.

With respect to Figure 2A, each of the decoding sections 35 operate to reverse the above-described encoding steps, except that the bit allocation decisions have already been made, and some of the packets may have been lost during transit. In particular, in steps 52 and 53, the decoding section 35 first reads the surviving packets and sorts those packets into parity groups. If any parity groups have one or missing members, section 35 reconstructs the missing member in step 54.

In step 55, each decoding section 35 decodes all of the data blocks into their respective subband bitplanes. If a high-order bitplane block is missing, then all of the lower order bitplane blocks corresponding to the high order block are not decodable. Finally, in step 56, the image is reconstructed: missing coarse band pixels are reconstructed by averaging neighbors, while missing detail band pixels are reconstructed using the subband mean. In Figure 2A, the reconstructed image 14c is shown on computer display 60 of client station 62 with image features as shown in image 14 of Figure 1.

In a flow chart format, Figure 2B illustrates alternative encoding methodology according to the invention. In step 40', a wavelet subband decomposition is applied to the image 14'. In a complete wavelet

decomposition, the coarse-scale subband is a single pixel value corresponding to a weighted average of all pixels in the image 14'. Because certain header information is maintained with each subband, it is more efficient to stop the wavelet transformation at some point short of a single pixel, e.g., a 32 x 32 coarse-scale image, and to transmit this image untransformed. This base image is referred to herein as the "coarse scale subband," similar to the DC band of a JPEG image. The detail (non-DC) bands are thus refinements of the image, whereby each successive band provides the information necessary to double the image resolution. Image 14a' illustrates the decomposition of image 14'.

In step 42', quantizer redundancies and parity levels are assigned, for example as described in connection with the Bit Allocation for Source Coding Sections 18, 18' of Figures 1 and 1A.

In step 44', each band is distributed across many packets by interleaving pixels so that a lost packet will not cause a catastrophic band loss. For example, Turner et al., Image transfer: an end-to-end design, SigComm 92, 258-268, describes one such suitable interleaving scheme. Distributing subbands does not reduce expected distortion because the chance of some loss in a given subband is increased by distribution; but it does reduce the variance in the expected distortion by increasing the population subject to the transmission experiment.

The incentive for subdivision is tempered by the desired step of encoding a descriptive header within each independent image block. Since image transmission is lossy, and since only an unknown subset of network packets arrive intact, a header which describes enough detail so as to permit lossy reconstruction of the block's subband is added to each block. For example, an image block of up to about 150 bytes provides suitable subdivision.

Likewise, header information can be about 15 bytes per block, so that if many small subbands are not broken into blocks, roughly 10% of the compressed image ends up as header information.

Alternatively, those skilled in the art should appreciate that the header could be located in a few heavily shielded packets, providing a more efficient configuration since much of the header information is replicated between blocks from the same subband.

In step 46', the wavelet coefficients are compressed within the relevant block. In one embodiment of the invention, these wavelet coefficients are compressed using adaptive arithmetic coding. Arithmetic coders emit $\Sigma - \log_2 p_i$ bits where p_i is the predicted probability of the ith event. In adaptive coding, the relative frequencies of past events are remembered in histograms and are used to estimate the probability of future events for the purposes of coding. To ensure that no event has a predicted probability of zero, histograms are usually initialized so that all possible events have a frequency of one. The histogram is adapted to the actual frequencies encountered as the input is read. For a large dataset, the inertia represented by the initial flat histogram is relatively unimportant.

The amount of time available for the histogram to adapt to the input distribution is reduced through subdivision into blocks. To compensate for this effect, the FLIIT methodology of this embodiment preferably incorporates step 48'. Specifically, step 48' utilizes the following two-histogram scheme, which adapts much more quickly than a single histogram scheme:

- Initialize two histograms, one histogram (F) that is flat with every possible value initialized to one, and one histogram (H) that is empty

with a single symbol, e.g., the escape symbol with an initial probability and frequency of one.

- Whenever an input symbol appears with non-zero probability (frequency) in histogram H, code the input symbol within histogram H and increment its frequency therein.

- Whenever an input symbol appears with zero probability in histogram H, code the escape symbol within histogram H, and code the input symbol within histogram F. This new symbol is added to histogram H with a frequency of one, and histogram F is never again used to code this symbol.

In step 50', after the compressed blocks are generated, redundancy is added. First, the blocks are sorted by protection level and size, in order of decreasing protection and size. Blocks requiring replication are replicated; and a given block and its replicas are all assigned the same parity group number, which prevents them from being included in the same network packet.

Blocks requiring the same level of parity protection are preferably grouped together by type, even though it is not generally possible to meet this preference exactly. For example, four data blocks can exist which are preferably arranged in a group of five data blocks protected by a parity block. In such a case, one block with less stringent protection requirements is promoted, if available, to round out the group. The promotion of this block is more efficient than the alternative, which leaves parity groups unfilled, and which effectively promotes all of the members of a group to a higher level of protection.

Sorting by size also helps to keep similarly-sized blocks in the parity group, which is valuable because the parity block should be as large as the largest data block in the group.

In step 51', the data and parity blocks are readied for transmission. When the network 12' is congested, throughput will be gated by router scheduling, rather than by bandwidth. Routers schedule communication channels using a round-robin-type algorithm which is insensitive to packet size. Accordingly, users of packets which are smaller than the largest packet transmitted by the network will pay a throughput penalty. Conversely, users who send overly-large packets that become fragmented en-route lose an entire packet whenever a single fragment is lost, also resulting in reduced throughput.

Because the largest Internet packet size which is guaranteed transmission on the Internet 12' without fragmentation is 576 bytes, the invention preferably groups information into packet sizes which are 576 bytes. Those skilled in the art should appreciate that differing network packet sizes can be used with the invention and that the 576-byte network packet size is subject to change with the growth and expected protocol changes of the Internet 12'. By way of example, one suitable packet packaging includes a largest-first first-fit heuristic protocol to pack blocks into 550-byte UDP packets. ATM further incorporates data blocks of 58 bytes; and this size is also suitable for the invention. If desired, the protocols of the invention can include additional restrictions: that blocks from the same parity group are not allowed in the same packet, and that blocks from the same band are not allowed in the same packet.

Rate Control Section

Because the Internet is a shared medium, system 10 of Figure 1 preferably controls the rate at which data is transmitted thereon or risks causing congestion in the network, resulting in lost packets and reduced performance for every connected user. In the prior art, TCP protocol implemented in the Reno release of BSD UNIX, known to those skilled in the art, controls its rate by starting out very slowly, by slowing down when packets are dropped, indicating congestion, and by speeding up otherwise. The problem with the TCP Reno strategy is that it simultaneously induces a certain level of packet losses on the Internet.

A second prior art rate control method implemented in TCP Vegas, see, e.g., L. S. Brakmo et al., TCP Vegas: New techniques for congestion detection and avoidance, Proceedings of the SIGCOMM '94 Symposium (1994), operates to compare expected throughput rates with actual throughput rates. Whenever the rate of packet reception drops below the rate of packet transmission, the network must be storing or dropping the excess data. Accordingly, TCP Vegas does not require packet losses in order to function, and typically delivers higher throughput than TCP without degrading the performance of other TCP connections.

Neither of the prior art TCP Reno or TCP Vegas rate control schemes are appropriate for the transmission of compressed images across the Internet. They are not appropriate because compressed images are much smaller than the dataset size required to achieve steady state transmission. By way of example, because TCP Reno overshoots the channel's actual throughput by a factor of two at the end of a slow start-up, a delay of 2.5 seconds before steady state can be measured as well as heavy packet losses at the end of start-up. See

L. S. Brakmo et al., TCP Vegas: New techniques for congestion detection and avoidance, Proceedings of the SIGCOMM '94 Symposium (1994).

In accord with the invention, FLIIT methodology preferably implements rate control using a delay-based scheme, whereby FLIIT clients, e.g., any of the users 30, 34a, 34b, sends an acknowledgment along Internet 12 to the Rate Control Section 22 every 16th packet. Rate Control Section 22 uses this acknowledgment to measure the round trip time ("RTT") and the current loss rate. The smallest observed RTT is the base round trip time ("BRTT"), which is assumed to be the round trip time in an uncongested network and which is generally the fastest round trip travel time. System 10 attempts to keep the actual RTT just above the BRTT by adjusting the sending rate. In other words, system 10 attempts to keep a small constant number of packets stored in the network which are ready in case congestion drops; yet there are not so many stored packets that they contribute to losses.

On the Internet, the number of connected networks relates to the "packet store," which is the number of packets in the Internet routers. Figure 1C shows a typical transmission of a packet 114 through various Internet routers 112. The routers 112 act as FIFOs because the first packet 116 within each packet store 118 is transmitted to the next location, which could be a another router or the end recipient. One goal of the Rate Control Section, therefore, is to ensure that some packets, but not too many, are transmitted to and stored within the routers 112. It adjusts the transmission rate so as to select the number of packets within the routers, to maximize image quality as a function of time.

The packet store S corresponding to a given RTT is thus the number of packets sent during that interval. The extra packet store $\Delta S = S$ (RTT -

BRTT)/RTT, and system 10 operates to keep ΔS in the range $\Delta S_L < \Delta S < \Delta S_H$, where ΔS_H and ΔS_L are determined empirically. ΔS_H is the respective router's estimate of how large the packet store can be above the first packet, i.e., those packets in the store 118 above the packet 116, Figure 1C. ΔS_H should be set low enough so that system 10 does not over-drive the network; and ΔS_L should be set sufficiently high so that system 10 responds to available network bandwidth.

Given S, ΔS_H, and ΔS_L, system 10 adjusts the interpacket sending interval I = RTT/S using a method such as Newton's iterative method. If, for example, $\Delta S < \Delta S_L$, system 10 can increase the sending rate to $I_{new} = I - I (\Delta S_L - \Delta S)/S$. If, on the other hand, $\Delta S > \Delta S_H$, system 10 can decrease the sending rate to $I_{new} = I + I (\Delta S - \Delta S_H)/S$.

System 10 can also function to react to systematic packet losses, while ignoring sporadic packet losses. By way of example, if losses over some threshold T_1, between acknowledgments, cause a small reduction in the sending rate, I_{new} can be set to $I_{old} F$, where F is a constant less than one (e.g., 0.9), to slow down the response. If on the other hand, losses over a larger threshold $L > T_2$ cause a larger reduction in the sending rate, then I_{new} can be set to $I_{old} f$, where f is a constant smaller than F (e.g., 0.5).

Those skilled in the art should appreciate that other rate control schemes can be used with the invention. For example, an off-line process can be used to select a transmission rate for FLIIT packets, such as by picking the knee on the network's load/loss curve. *See, e.g.,* C.L. Williamson et al., Loss-load curves: Support for rate-based congestion control in high-speed datagram networks, Proceedings of SIGCOMM 91, pp. 17-287 (1991). In accord with the invention, streams of packets containing roughly 550 bytes were sent at various

transmission rates, and the loss rate was measured for each rate. As shown in Figure 2C, the loss rate as a function of transmission rate was relatively constant at rates below about 4ms-per-packet, as compared to higher rates. Each curve of Figure 2C corresponds to a different transmission rate. Above 4ms-per-packet, however, the loss rate increased sharply, relative to rates corresponding to 4ms-per-packet and below. To avoid high loss rates, and to avoid impacting other applications operating on the Internet, a reasonable transmission packet rate is therefore 4ms-per-packet, in accord with the invention. Faster transmission rates can be chosen at other times. For example, rates of 2ms-per-packet are effective between about 03:00 and 04:00 EDT; but such rates generate high losses when the Internet becomes busy during the day. During daylight hours, the loss rate never drops much below about 5% regardless of the transmission rate.

The rate control section of the invention thus provides certain advantages over the art. For example, because the buffer capacity of the Internet between any two well-separated nodes is typically greater than the size of a well-compressed image, a server could transmit an entire image in less than one RTT. This however is not feasible with TCP because TCP has a slow start-up time for each connection; and further takes many round trips to reach full speed. This is fine for megabyte transfers, but inappropriate for smaller and widely used image sizes, e.g., 8-kilobyte images. In a preferred embodiment of the invention, therefore, the FLIIT server remembers the effective transfer rates across its active connections, and effectively removes the slow startup as an issue

Stopping Criterion

In certain instances, FLIIT packets may be lost or delayed for long periods of time. If system 10 (Figure 1) waits too long for slow packets, system 10 loses responsiveness. If, on the other hand, system 10 does not wait long enough, it loses packet data. This tradeoff, between transmission speed and packet loss, has a practical solution, in accord with the invention. Specifically, in a preferred embodiment, system 10 incorporates the tradeoff into the resource allocation algorithm to choose an optimal time to stop waiting for lost packets. Typically, the ideal stopping point is the expected time of arrival of the last packet plus the standard deviation of the interpacket arrival time.

Other stopping criterion can be used with the invention. For example, in one embodiment of the invention, system 10 of Figure 1 sends packets at a constant rate, which preferably keeps Internet congestion down. If, for example, a packet is sent every b time units, where b is less than or equal to the throughput of the network, and the network delivers all packets after a fixed time delay, then the n -th packet will arrive at time $T_n = a + (n - 1)b.$, where a is the arrival time of the first packet. On the Internet 32, packets are delayed by variable lengths of time. System 10 can incorporate this variability into the arrival time model by adding a random delay variable X_n. Accordingly, $T_n = a + b(n - 1) + X_n$. System 10 then determines a stopping time T_{stop} after which it stops waiting for packets to reconstruct the image 14. Packets arriving after time T_{stop} are then considered lost by system 10.

Given the distributions of X_n, the probability $P(T_n > T_{stop})$ that packet n will be lost due to excessive delay can be determined. By randomizing the order of the packets transmitted, the probability of any given packet being the n -th

packet transmitted is 1/N , where N is the total number of packets sent. The probability of a particular packet being lost due to delay is

$$p_{delay}(T_{stop}) = \frac{1}{N}\sum_{k=1}^{N} P(T_n > T_{stop})$$

The overall probability that a packet is lost is then $p_{loss} = 1 - (1 - p_{drop})(1 - p_{delay})$, where p_{drop} is the probability of the packet being dropped in transit.

The stopping time affects the loss rate observed by the receiver, e.g., any of the units 34a on the Internet 32. The reconstructed image distortion is thus a function of the number of data, the redundancy bits, and the stopping time. Because the constraint is on the number of bits sent, and not on the length of time required to receive the image, the optimal value of T_{stop} is infinity. If the goal is to maximize responsiveness, the time required to receive the image is constrained rather than the total number of bits sent. This can be done by setting the cost function as the sum of the time required to send the bits in the image plus the waiting time. This results in a new set of cost and distortion functions which depend on the bit allocations as well as the stopping time. By varying the stopping time in the allocation algorithm, discussed above, the bit allocations and stopping time can be obtained in an optimized fashion.

Figure 2D illustrates observed and fitted cumulative density functions for the packet delays X_n, which correspond to a set of independent, identically distributed Poisson random variables with parameter λ. The data was gathered from ten 160-packet transmissions. Through the method of moments, the offset rate, a, and the sending rate, b, can determined by least squares and the parameter λ to isolate the delay X_n. The resulting delay is normalized to have

mean 0 and variance 1. The superimposed solid curve is the cumulative density function for an equivalently normalized Poisson random variable.

As illustrated in Figure 2D, the stopping time model of this embodiment describes the distribution of delays. The server can update its knowledge of network conditions by periodically obtaining these quantities from the receiver. The typical stopping time is the expected time of arrival of the last packet , $a + b(N - 1) + \lambda$, plus a delay ranging from 0 to the square root of λ, which is the standard deviation of the delay.

Experimental Results

(1) Experiment 1.

Using the well known Lena image at 256 x 256 resolution, we generated sets of packets using the FLIIT methodology discussed herein, as well as three different fixed-parity schemes, such as shown in Figure 3. The fixed-parity schemes used the same bit allocation as in FLIIT in order to determine quantizer resolutions for each subband, but no adaptive-coding was done for the parity bits. Experiment 1 therefore shows only the effects of adaptive versus fixed distribution of redundancy. In the fixed parity 3 scheme, each data block was replicated three times. In the fixed parity 1/3 scheme, groups of three data blocks were protected by a single parity block. In the fixed parity 0 scheme, no parity blocks (redundancy) were used. The "Y" axis of Figure 3 represents the peak signal to noise ration (PSNR), a logarithmic indicator of image quality; and the "X" axis represents the expected loss rate. As shown, the FLIIT methodology of the invention dominates the other schemes, usually by several dB.

In Experiment 1, packets were generated within each scheme using 8:1 compression, and with expected loss rates ranging from 0% to 50%. For each combination of parity scheme, compression ratio, and loss rate, we ran simulated transmission experiments in which packets were deleted by subjecting each packet to an independent pseudo-random Bernoulli trial. Images were then reconstructed from the remaining packets, allowing image comparisons and calculations of actual image distortions.

Figures 4A, 4B, 4C illustrate the fixed parity transmission results of Experiment 1. Figures 5A, 5B and 5C illustrate Lena image transmissions under FLIIT testing of Experiment 1. Compared to the loss rates tested, the FLIIT methodology of the invention has the overall best performance. The fixed parity 3 scheme performs best for high loss rates because of the large amounts of transmitted redundancy. At high loss rates, FLIIT also uses large amounts of redundancy, but it distributes these redundancy bits more selectively than the fixed scheme. In particular, FLIIT methodology shields the low-frequency portions of the image since the loss of a low frequency data block results in a much larger error than the loss of a high frequency block. The extra shielding is also relatively inexpensive, since there are relatively few low frequency coefficients.

More particularly, Figures 4A-4C and 5A-5C show, respectively, the effects of compression and transmission losses on the 256 x 256 Lena image under the fixed parity 3 scheme and under FLIIT. The images have been compressed from 64K to 9.5K (8:1 compression plus a roughly 20% packet header overhead cost), including the parity blocks, and all packets have a 50% probability of being lost. In effect, these images have been reconstructed from 4K of randomly selected data. These data show that transmissions with FLIIT

methodology perform well even at very high error rates. Figures 4A and 5A have a 90th percentile image quality; Figures 4B and 5B have a 50th percentile image quality; and Figures 4C and 5C have a 10th percentile reconstructed image quality. These figures represent a very severe test as images are reduced to roughly 9Kbytes (18-20 packets with overhead) and then packets are randomly eliminated in independent trials, so that well under 50% of the packets typically survive.

Figure 5D illustrates the Lena image of Figures 4 and 5 transmitted via TCP/IP. In comparison, the FLIIT transmitted images of Figures 5A-5C required between 0.8 to 2.0 seconds to transmit, while the TCP/IP transmitted image typically required between 1.4 and 12.3 seconds to transmit. On average, 11.5% of Internet packets were lost during each transmission of Figures 5A-5C. FLIIT thus trades variability in image delivery time for variability in image quality.

(2) Experiment 2.

In Experiment 2, we measured image quality (PSNR) as a function of transmission time for both FLIIT and TCP. Time begins when a client requests an image, and ends when the client decides that it has received an image. We did not include the decode time, which is the same for both clients, and which is practically negligible. For FLIIT protocol, the client makes its request with a single UDP packet. For TCP transport, the client makes its request over a TCP connection. FLIIT images are returned using UDP. TCP images are returned using TCP. The TCP images were generated using the same compression routines as the FLIIT images, but there was no redundancy or blocking, eliminating all of the over-head which FLIIT needs for reconstruction after lossy transmission, but which are unnecessary after lossless transmission.

Accordingly, the prior art TCP method was not burdened or handicapped with the overhead byte requirements of FLIIT.

In accord with the invention, discussed above, the FLIIT client calculates a running estimate of the expected time of arrival of the last packet. The client waits some period beyond this time, typically one standard deviation of the interpacket arrival time, and decodes the image. The exact amount of extra time to wait is calculated and specified by the FLIIT server. The TCP client stops when the complete image has arrived.

We used a real Internet connection for this experiment. The connection was between Dartmouth College in Hanover New Hampshire, and Stanford University in Stanford California. The participating computers were separated by 20 hops. For convenience, we ran the client and the server locally, but sent the data across the continent, by routing network packets from our local client to a local pseudo server, which bounced these packets off of the Stanford machine's echo server, and forwarding the returning packets to our local server. Traffic from the server to the client was also similarly redirected through the remote echo server. The transport methodologies used during experimentation are illustrated in Figure 6. Data packets for FLIIT methodology originated from a local FLIIT user 70 and through the forwarding server 72 at Dartmouth College. Stream data for TCP originated at a local TCP user 74 and similarly forwarded through server 72. The data from either transport methodology was thereafter forwarded through an echo server 76 at Stanford University, and then back to the respective servers 70, 74.

The image used in Experiment 2 was again the Lena image at 256 x 256 resolution. We transmitted Lena at different compression ratios, 160 times for each sample. We ran the experiment under two different sets of circumstances:

daytime and nighttime, both on weekdays. Daytime was 12:00-18:OOEDT. Nighttime was 02:00-08:OOEDT. We set the expected loss rate, expected packet arrival rate, and standard deviation of interpacket delay to 1.3%, 4.4ms per packet, and 10.4ms for the night experiments, and 8.2%, 4.6ms per packet, and 12.3ms during the day.

The results of this experiment are graphed in Figure 7, which plots image quality in terms of PSNR, a logarithmic function of the mean squared error, and as a function of transmission time. Plotted points correspond to median values, while error bars indicate first and third quartiles. TCP curves have error bars only in the time dimension because they deliver consistent quality. FLIIT has error bars in both dimensions because both quality and time are variable. For equivalent quality, FLIIT methodology is roughly twice as fast as TCP at night, and four times faster than TCP during the day. FLIIT has minimal variation in transmission time, while TCP transmission times vary widely, especially during the day.

FLIIT methodology, in accord with the invention, uniformly outperformed TCP for equivalent image quality. Highly compressed FLIIT images were transmitted over twice as fast as their TCP counterparts, presumably because fewer round trips are necessary to establish a FLIIT connection. Moderately compressed images were transmitted more quickly because FLIIT does not retransmit dropped packets or wait multiple round trip times for the last few packets. During the day, when the Internet is congested, FLIIT methodology is more than four times faster than TCP, even for high quality images.

In summary of Experiment 2, FLIIT accepts some variance in quality for a large improvement in throughput and a large reduction of the multisecond

variance in time accepted by TCP. Although TCP makes the right tradeoff for applications requiring perfect transmission, FLIIT methodology, in accord with the invention, makes the right tradeoff for interactive and real-time applications. Other experimentation with wavelet-based coder transforms, according to the invention, has yielded PSNR's for the 512 x 512 Lena image of image within 0.3 to 0.9 dB of images created by very high quality prior art coders such as described by J.D. Villasenor et al., IEEE Trans. Image Processing (1995).

(3) Experiment 3.

In a third experiment, a comparison was made between the impact of a FLIIT operation on a TCP session and the impact of a TCP operation on a TCP session to determine whether FLIIT methodology over-utilizes the Internet as compared to TCP. Because of a possible concern that some of FLIIT's performance might derive at the expense of other network clients, e.g., that FLIIT methodology might appropriate disproportionate bandwidth away from TCP connections, Experiment 3 was conceived to measure, separately, the performance of (1) FLIIT alone, (2) TCP alone, (3) FLIIT with FLIIT, (4) TCP with TCP, and (5) FLIIT with TCP. The format of the experiment was otherwise the same as the Experiment 1, e.g., same image, same network, same number of trials, etc.

In the paired methods of Experiment 3, two images were transferred together, and were allowed to finish separately. Because FLIIT image transfers always finish much sooner than TCP, the TCP vs. FLIIT experiments show the performance of TCP running by itself for much of the time. This experiment particularly measures the impact on the network of FLIIT vs. TCP protocols performing the same task.

Figures 8A, 8B illustrate the results of Experiment 3. Each graph plots image quality as a function of time. Figure 8A shows the performance of FLIIT by itself, FLIIT competing for bandwidth with a TCP connection, and FLIIT competing for bandwidth with a FLIIT connection. In Figure 8B, analogous graphs feature TCP performances. Because Experiment 3 was carried out on the Internet, which has many other users, there is no expectation that two simultaneous transmissions should take twice as long as one transmission.

Even though FLIIT transmission were generally faster, the two FLIIT connections of Figure 8A degraded each other's performance more than the two TCP connections of Figure 8B. This indicates that FLIIT utilizes a larger fraction of the aggregate network bandwidth than TCP, so that when two FLIIT connections run together, they have a larger effect on each other.

For low quality image transmissions, TCP runs about as fast competing with FLIIT as it does competing with TCP. For medium quality images, TCP runs slightly slower with FLIIT as compared to running with TCP. At the highest quality, TCP runs faster with FLIIT than with TCP. This indicates that transferring a high quality image with FLIIT has less effect on the network than transferring a high quality image with TCP, but that while FLIIT is transferring an image, it has a greater effect on the network than does TCP.

The invention thus demonstrates a system which combines source and channel coding to produce an image transfer protocol that transfers images of a given quality twice as fast as the TCP protocol at night, and four times faster than TCP during the day. Note that this figure is comparing wavelets to wavelets. Further, FLIIT will outperform JPEG image transmission by an even greater margin, since JPEG images are larger than wavelet images.

The FLIIT methodology presented herein is particularly appropriate for image previewing, progressive image transmission, transmission of moving pictures, and broadcast applications.

Those skilled in the art should appreciate that changes can be made within the description above without departing from the scope of the invention. For example, it should be apparent that image compression within the Image Compression Section 16, Figure 1, can utilize DCT-based schemes such as JPEG by replacing wavelet subbands, described above, with blocks of DCT coefficients of comparable frequencies.

The invention thus attains the objects set forth above, among those apparent from preceding description. Since certain changes may be made in the above apparatus and methods without departing from the scope of the invention, it is intended that all matter contained in the above description or shown in the accompanying drawing be interpreted as illustrative and not in a limiting sense. It is also to be understood that the following claims are to cover all generic and specific features of the invention described herein, and all statements of the scope of the invention which, as a matter of language, might be said to fall there between.

Having described the invention, what is claimed as new and secured by Letters Patent is:

1. A method of decomposing a digitized signal into a collection of subbands for transmission over the Internet, comprising the steps of (a) allocating forward error correction bits and quantizer precision to each subband in order to reduce expected image distortion subject to an overall bit budget; and (b) transmitting the quantized image and forward error correction bits over the Internet.

2. A method according to claim 1, wherein the step of allocating forward error correction bits and quantizer precision to each subband comprises the step of minimizing expected image distortion subject to the bit budget.

3. A method according to claim 1, wherein the step of allocating forward error correction bits and quantizer precision to each subband occurs automatically.

4. A method according to claim 1, wherein the step of allocating forward error correction bits comprises the step of decomposing the subband into outputs of quantizers.

5. A method according to claim 4, wherein the step of allocating forward error correction bits comprises the step of allocating forward error correction bits to each of the outputs.

6. A method according to claim 4, wherein the step of decomposing the subband into outputs of quantizers comprises decomposing the subband into outputs of nested quantizers.

7. A method according to claim 6, wherein one or more of the outputs comprise a succession of discrete refinements to a discrete representation of a real value.

8. A method according to claim 6, comprising the further step of entropy coding the outputs.

9. A method according to claim 8, wherein the entropy coding is selected from the group of adaptive arithmetic coders, static arithmetic coders, Huffman coders, and dictionary-based coders.

10. A method according to claim 9, wherein the dictionary based coders comprise one of Lempel-Ziv 77 coder and Lempel-Ziv 78 coder.

11. A method according to claim 4, wherein one or more outputs comprise a discrete representation of a real value.

12. A method according to claim 11, wherein the real value comprises a subband coefficient from a subband decomposition of a digitized signal.

13. A method according to claim 4, wherein the step of decomposing the subband into outputs of quantizers comprises one of the following: a wavelet transform, a discrete cosine transform, and a motion compensation and discrete cosine based decomposition.

14. A method according to claim 13, wherein the discrete cosine transform comprises a JPEG discrete cosine transform.

15. A method according to claim 13, wherein the motion compensation and discrete cosine based decomposition comprise MPEG motion compensation and discrete cosine based decomposition.

16. A method according to claim 1, wherein the digitized signal is one or more of a digital image, a digitized audio, a digitized video, and a digitized geometric representation of an object.

17. A method according to claim 1, wherein the step of allocating forward error correction bits further comprises the step of modeling effects of network burst error to adjust the expected reconstructed image distortion.

18. A method according to claim 17, wherein the step of modeling effects of network burst error comprises the step of applying a Markov model to predict expected packet losses between two or more transmitted packets.

19. A method according to claim 17, further comprising the step of randomizing packet order in order to decorrelate burst losses during transmission.

20. A method according to claim 1, wherein the step of allocating forward error correction bits further comprises the step of allocating bits between tasks of encoding image subbands and protecting encoded data with forward error correction.

21. A method according to claim 20, wherein the step of allocating bits between tasks of encoding image subbands and protecting encoded data with forward error correction further comprises the step of determining a likely

distortion of the reconstructed image relative to compression and network losses, and subject to a total number of bytes transmitted.

22. A method according to claim 1, wherein the step of allocating forward error correction bits further comprises the step of allocating a fixed number of bits between redundancy and data depending upon an expected loss rate through the Internet.

23. A method according to claim 22, wherein the step of allocating a fixed number of bits further comprises the steps of shifting bits to redundancy for high loss rates, and shifting bits to data for lower loss rates.

24. A method according to claim 1, further comprising the step of storing final transmission rates for one or more connections to reduce subsequent startup time to any of the connections.

25. A method according to claim 1, further comprising the step of interleaving the subbands into smaller memory blocks.

26. A method according to claim 25, wherein the step of interleaving the subbands into smaller memory blocks further comprises the step of forming memory blocks of up to about 150 bytes.

27. A method according to claim 25, further comprising the step of compressing the memory blocks with an arithmetic decoder.

28. A system for transmitting a digital image over the Internet, the digital image of the type that includes a series of subbands, comprising: means for allocating forward error correction bits and quantizer precision to each subband

in order to reduce expected image distortion subject to an overall bit budget; and (b) means for transmitting the quantized image and forward error correction bits over the Internet.

29. A system according to claim 28, further comprising means for decomposing each of the subbands into outputs of quantizers.

30. A system according to claim 29, wherein the means for decomposing each of the subbands into outputs of quantizers comprises means for decomposing each of the subbands into outputs of nested quantizers.

31. A system according to claim 29, wherein the means for decomposing each of the subbands into outputs of quantizers comprises one or more of the following: means for performing wavelet transform, means for performing a discrete cosine transform, and means for performing a motion compensation and discrete cosine based decomposition.

32. A system for transmitting a digital image across the Internet, comprising: an image compression section for performing lossy image compression on the image; a bit allocation for source coding section for transforming the compressed image into a set of subbands, each subband being ranked relative to other subbands based upon its impact to image quality; a channel coding and expected image distortion section for allocating bits within the subbands between source and channel codes in order to minimize an expected reconstructed image distortion subject to an overall transmission bit budget; and means for transmitting a subset of the subbands with forward error correction bits on the Internet.

33. A system according to claim 32, wherein the image compression section comprises transform coder means for generating lossy compressed image.

34. A system according to claim 16, wherein the transform coder means comprises one of JPEG, wavelet, wavelet packet, and DCT transforms.

35. A system according to claim 32, wherein the image compression section comprises means for performing wavelet transforms.

36. A system according to claim 35, wherein the means for performing wavelet transforms further comprises (a) means for quantizing the electronic image by quantizing the coefficients using uniform quantizers, and (b) an arithmetic coder for coding resulting coefficients for entropy.

37. A system according to claim 32, wherein the bit allocation for source coding section comprises means for decomposing each of the subbands into outputs of quantizers.

38. A system according to claim 37, wherein the means for decomposing each of the subbands into outputs of quantizers comprises one or more of the following: means for performing wavelet transform, means for performing a discrete cosine transform, and means for performing a motion compensation and discrete cosine based decomposition.

39. A system according to claim 32, wherein the bit allocation for source coding section comprises means for dynamically allocating bits between source and channel codes depending upon conditions within a network.

40. A system according to claim 32, wherein the bit allocation for source coding section comprises means for finely quantizing a first group of coefficients, and coarsely quantizing a second group of coefficients, the first group having greater visual impact on image fidelity.

41. A system according to claim 32, wherein the bit allocation for source coding section comprises means for determining a quantization resolution for each subband based upon a trade-off between quantization error, thereby allocating quantizer resolution to obtain a minimum image distortion for a given bit expenditure.

42. A system according to claim 41, further comprising a rate control section having means for determining the bit expenditure based upon network conditions.

43. A system according to claim 32, wherein the bit allocation for source coding section comprises means for decoding the image into a series of nested quantizers.

44. A system according to claim 32, wherein the bit allocation for source coding section comprises means for determining which bitplanes to transmit and the redundancy applied to each bitplane.

45. A system according to claim 32, wherein the bit allocation for source coding section comprises means for assessing image distortion.

46. A system according to claim 45, wherein the means for assessing image distortion further comprises means for storing and assessing image distortion based upon a mean squared error function.

47. A system according to claim 32, wherein the bit allocation for source coding section comprises means for refining quantizer resolution based upon marginal analysis.

48. A system according to claim 32, wherein the channel coding and expected image distortion section comprises means for adaptively allocating quantizer resolution, thereby adding redundancy to transmitted images and optimizing a partition of bits between source and channel codes.

49. A system according to claim 32, wherein the channel coding and expected image distortion section comprises means for interleaving subbands into smaller memory blocks.

50. A system according to claim 49, wherein the means for interleaving subbands into smaller memory blocks comprises means for forming memory blocks of up to 150 bytes.

51. A system according to claim 32, wherein the channel coding and expected image distortion section comprises means for compressing subbands with an arithmetic coder.

52. A system according to claim 51, wherein the means for compressing subbands with an arithmetic coder comprises means for forming network packets having a fixed Internet size.

53. A system according to claim 52, wherein the fixed Internet size is 576 bytes.

54. A system according to claim 51, wherein the channel coding and expected image distortion section comprises means for adding redundancy to compressed bitplanes by adding parity bits to a bitstream.

55. A system according to claim 51, wherein the channel coding and expected image distortion section comprises means for interleaving bitplanes to reduce a visual impact of packet losses during transmission.

56. A system according to claim 51, wherein the channel coding and expected image distortion section comprises means for determining a probability of network packet loss.

57. A system according to claim 51, further comprising a rate control section for determining a round-trip travel time for transmission packets on the Internet.

58. A method for transmitting an electronic image between a first node and a second node, each node being connected to the other through the Internet, each node comprising a digital data processor, comprising the steps of: compressing the electronic image with forward error correction at the first node so that image bits are concentrated within image portions having greater visual impact; transmitting the compressed electronic image on the Internet and between the first node and second node; and reconstructing the transmitted image at the second node by rebuilding fragments lost during transmission according to the forward error correction applied to the image.

59. A method for transmitting an electronic image on the Internet, comprising the steps of: decomposing an electronically compressed image into

a series of subbands; selecting a subset of subbands to transmit on the Internet based upon a relative ranking between subands; allocating forward error correction bits to each subband within the subset in order to minimize an expected reconstructed image distortion subject to an overall transmission bit budget; and transmitting the subset and the forward error correction bits within packets on the Internet.

60. A method according to claim 59, further comprising the step of reconstructing the electronic image by (a) reading at least a portion of the packets from the Internet, and (b) decoding the subset of subbands according to the forward error correction bits allocated to the subbands, thereby rebuilding image fragments lost during transmission.

61. A method according to claim 59, further comprising the steps of assessing a waiting period for transmitted packets and terminating the waiting period at an approximate time corresponding to an expected arrival time of a last packet plus a standard deviation of an interpacket arrival time.

Figure 1

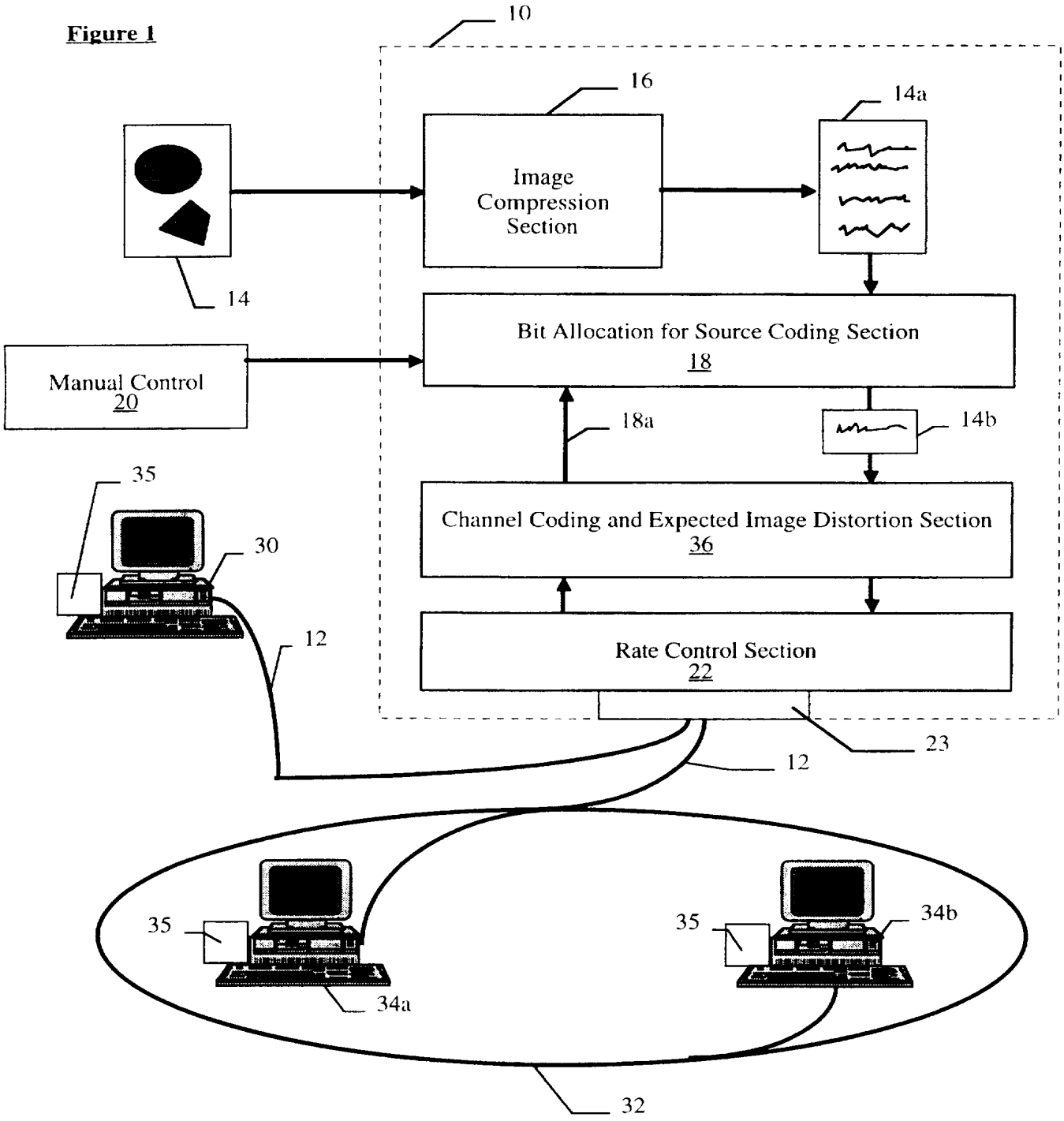

10
16
14a
Image Compression Section
14
Bit Allocation for Source Coding Section 18
Manual Control 20
18a
14b
35
30
Channel Coding and Expected Image Distortion Section 36
12
Rate Control Section 22
23
12
35
34b
35
34a
32

Figure 1A

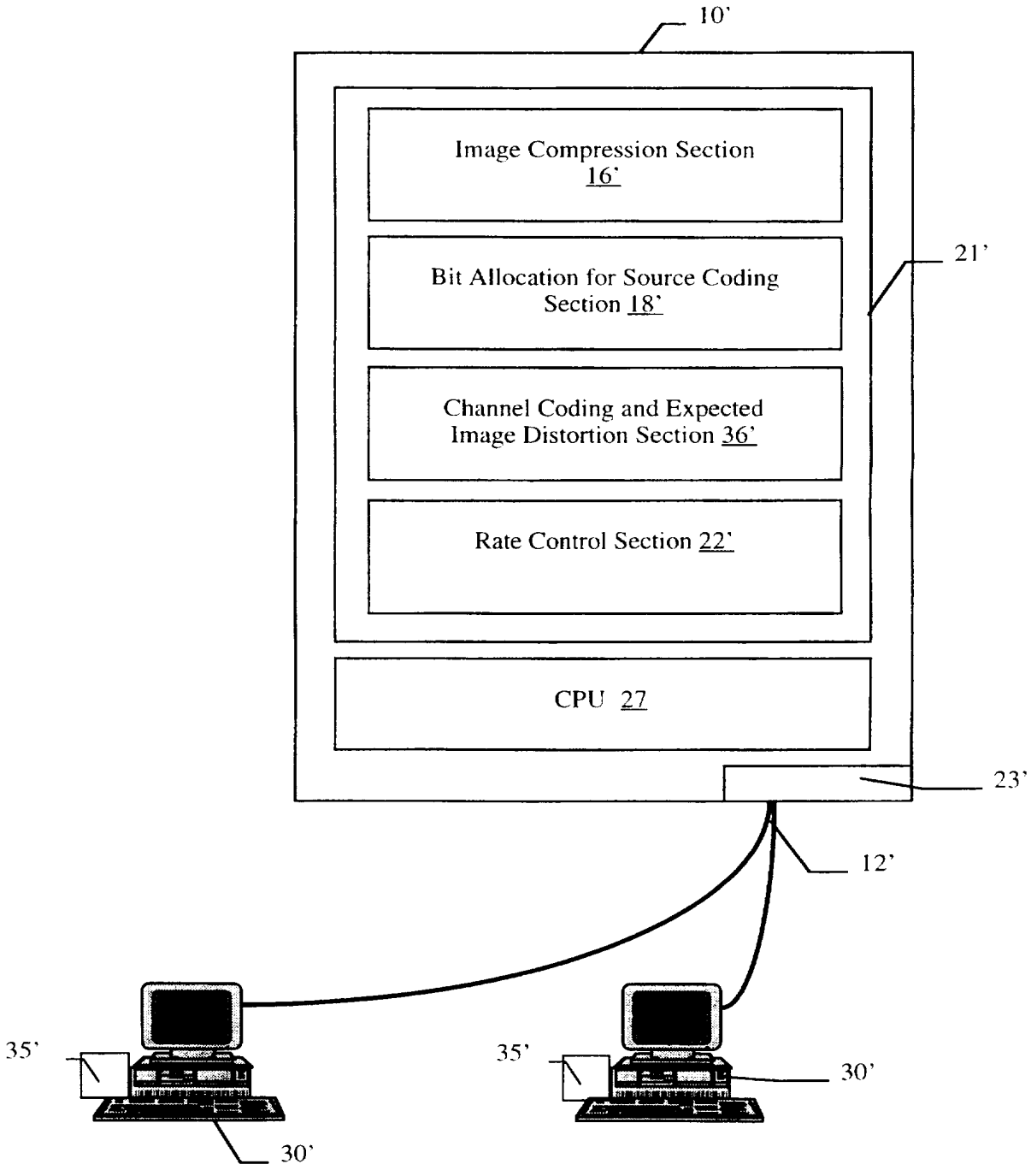

10'
Image Compression Section 16'
Bit Allocation for Source Coding Section 18'
Channel Coding and Expected Image Distortion Section 36'
Rate Control Section 22'
CPU 27
21'
23'
12'
35'
30'
35'
30'

Figure 1B

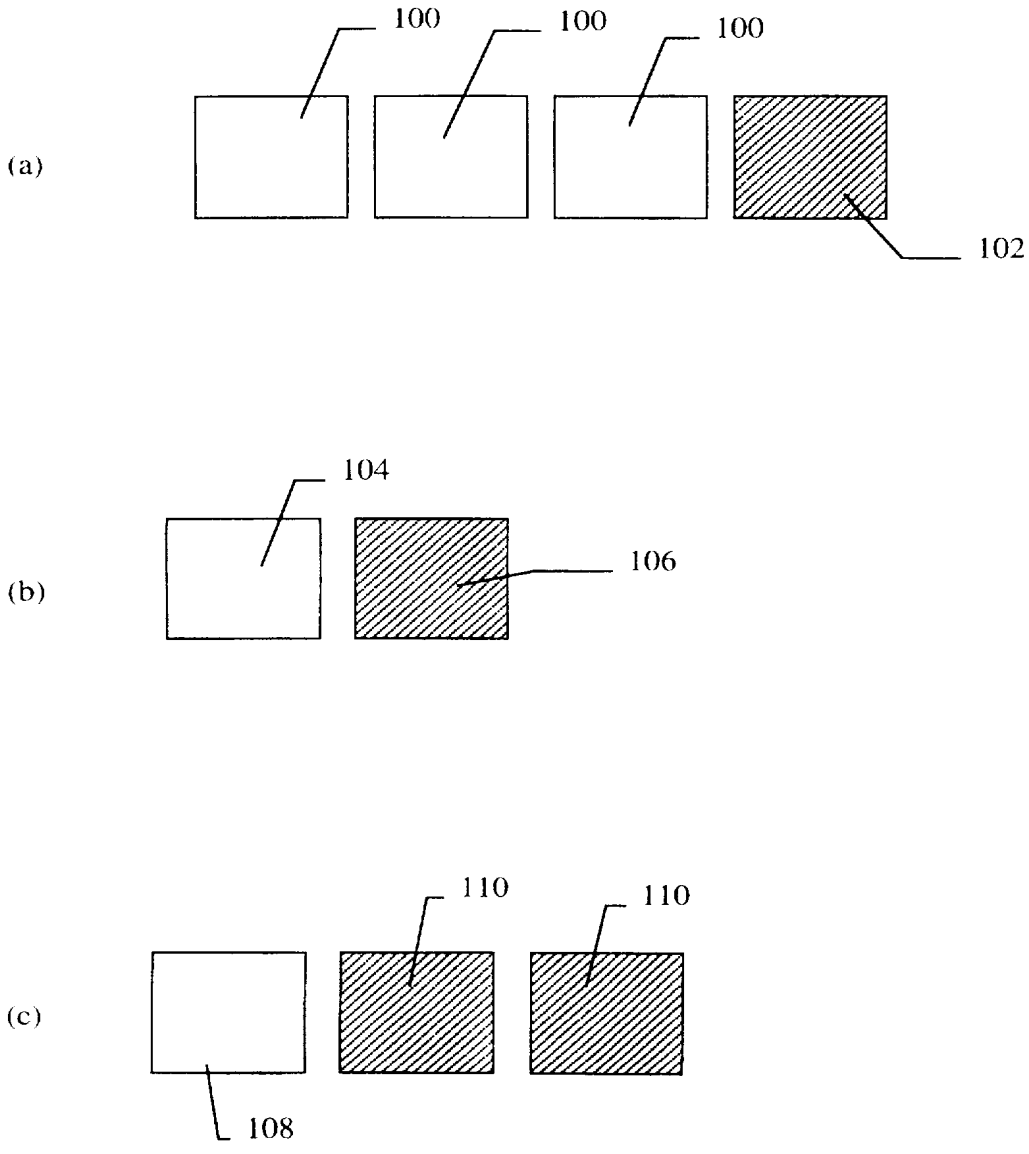


Figure 1C

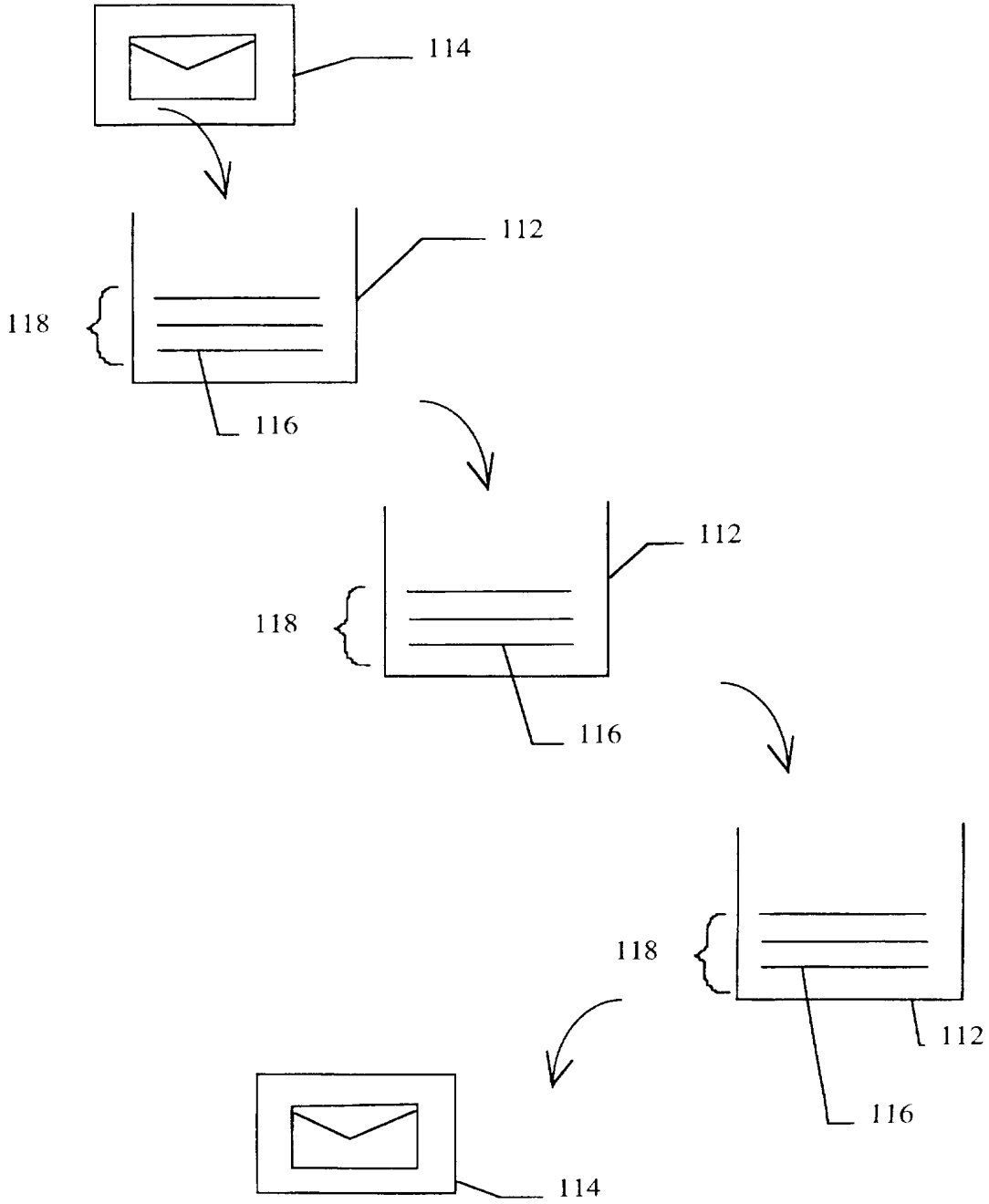
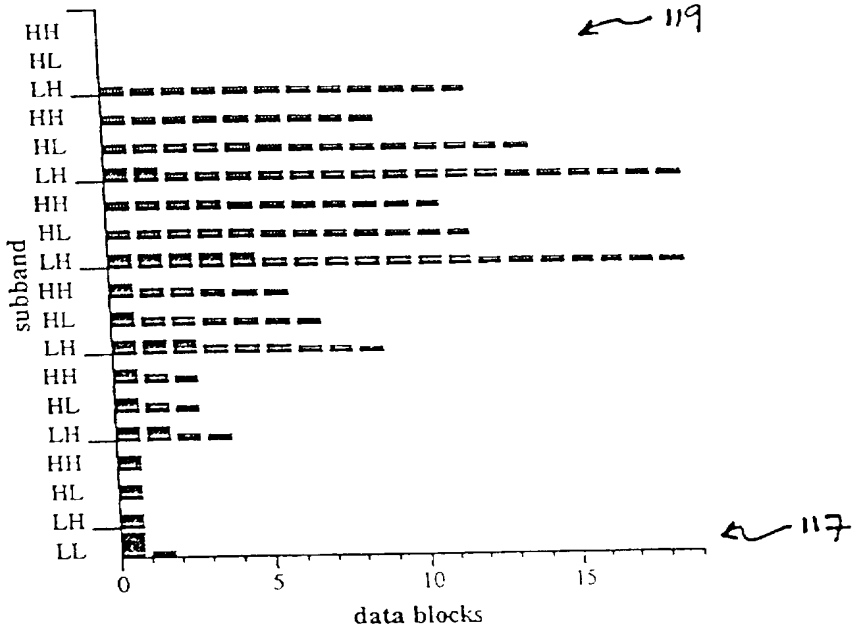


FIGURE 1D



Uneven Allocation of Forward Error Correction Bits

115

Figure 2

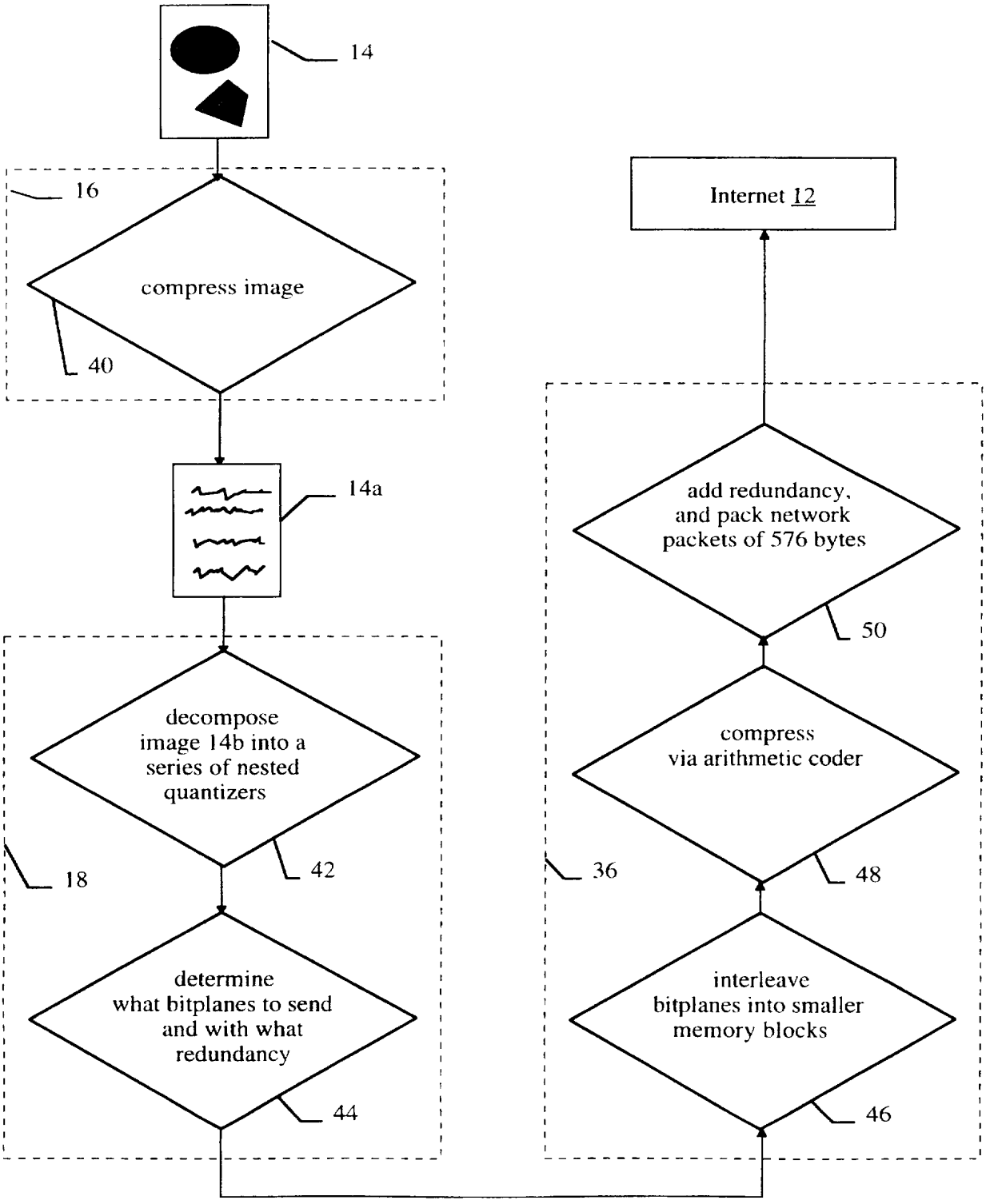

14
16
compress image
40
14a
decompose image 14b into a series of nested quantizers
42
18
determine what bitplanes to send and with what redundancy
44
Internet 12
add redundancy, and pack network packets of 576 bytes
50
compress via arithmetic coder
48
36
interleave bitplanes into smaller memory blocks
46

Figure 2a

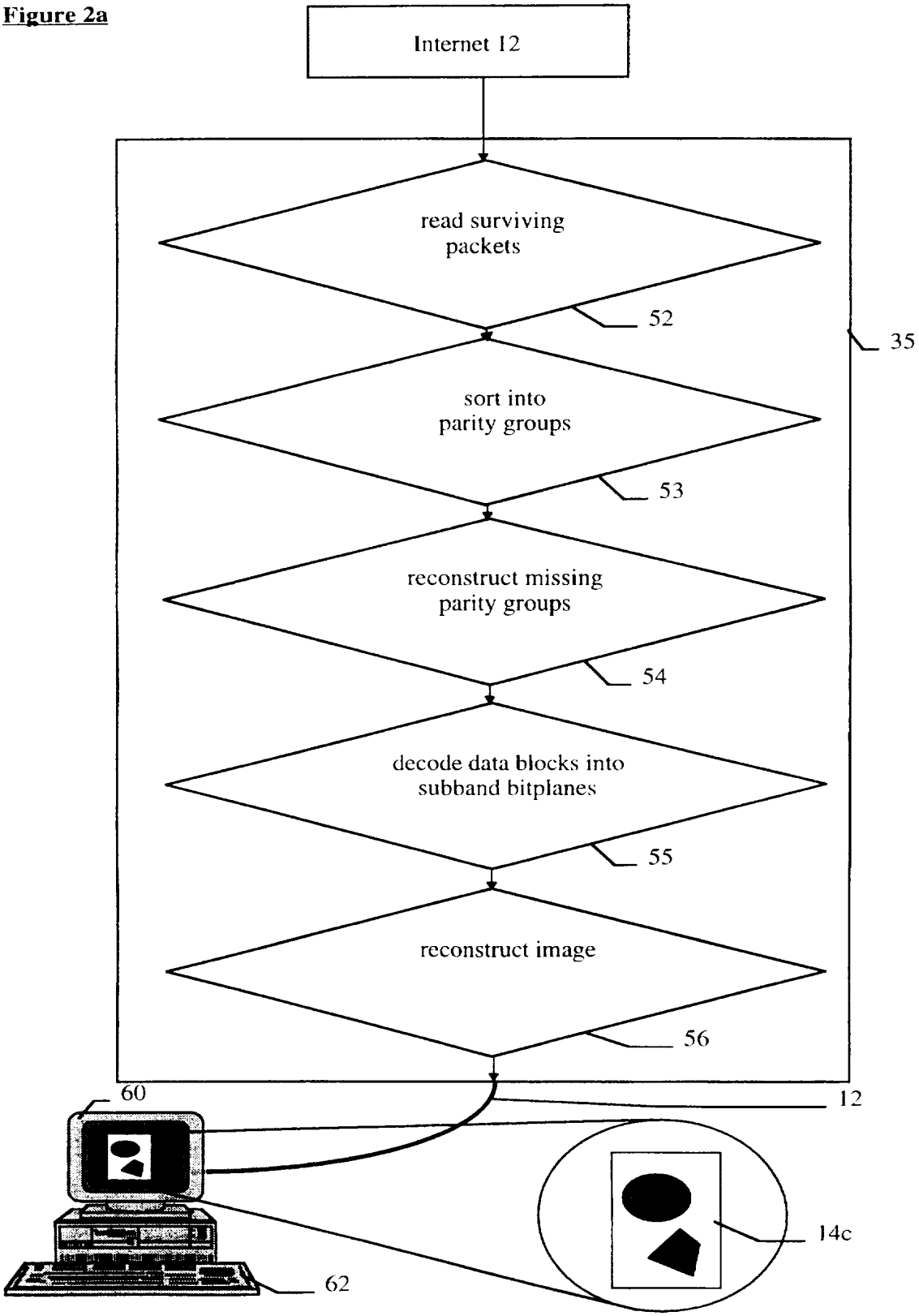

Internet 12
read surviving packets
52
35
sort into parity groups
53
reconstruct missing parity groups
54
decode data blocks into subband bitplanes
55
reconstruct image
56
60
12
14c
62

Figure 2B

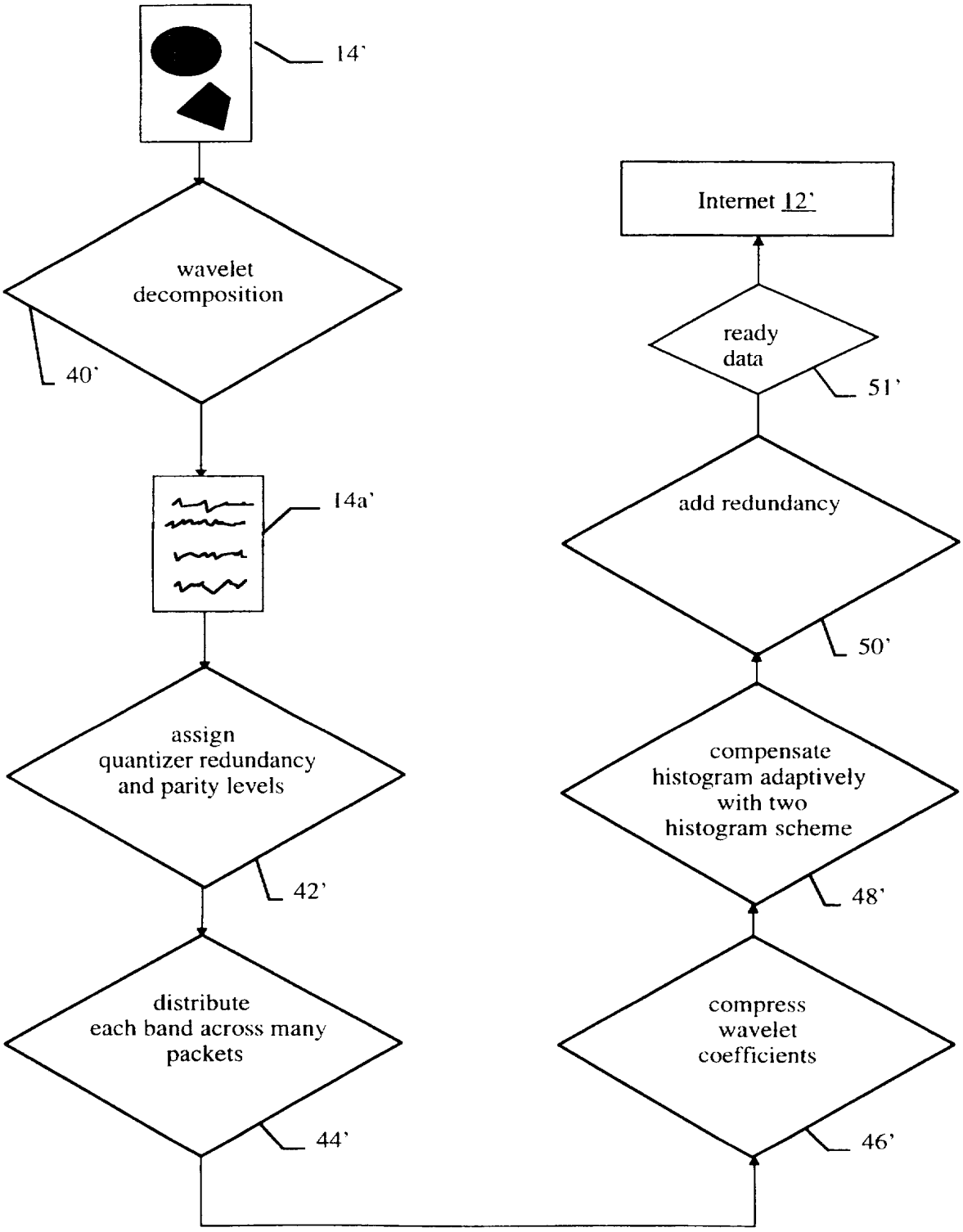

14'
wavelet decomposition
40'
14a'
assign quantizer redundancy and parity levels
42'
distribute each band across many packets
44'
Internet 12'
ready data
51'
add redundancy
50'
compensate histogram adaptively with two histogram scheme
48'
compress wavelet coefficients
46'

FIGURE 2C

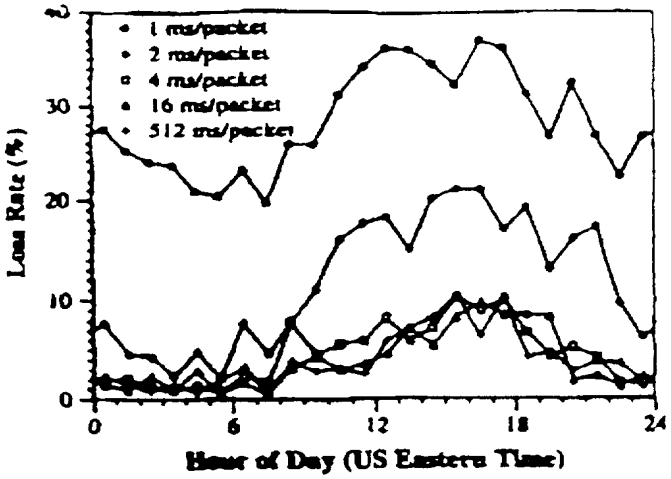
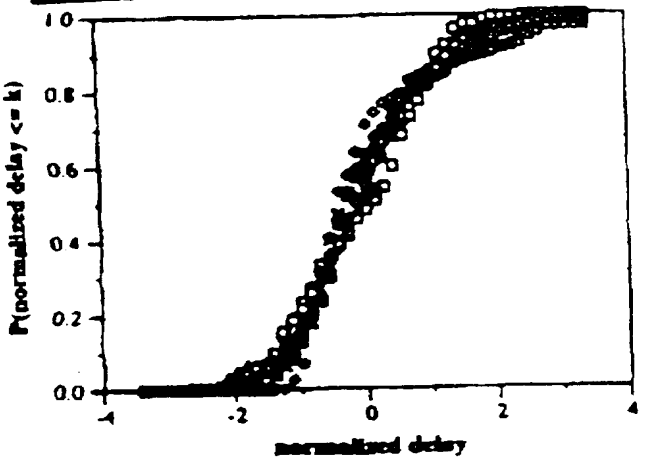
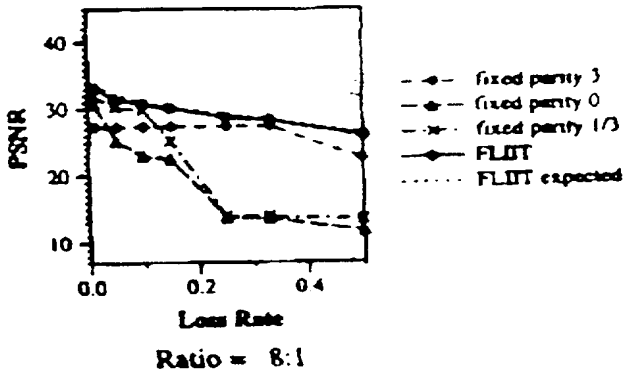


FIGURE 3

FIGURE 2D



Packet Delay Cumulative Density Function



Ratio = 8:1

Fig. 4A

Fig. 5A

FIG. 4B

FIG. 5B

FIG. 4C

FIG. 5C

Fig. 5D

FIGURE 6

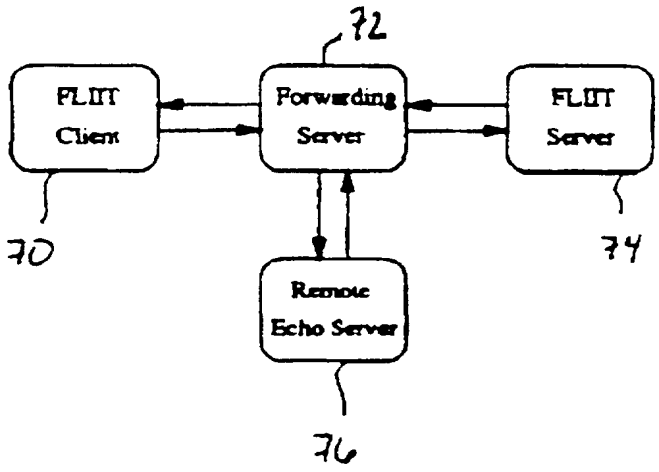

72
FLIT Client
Forwarding Server
FLIT Server
70
74
Remote Echo Server
76

FIGURE 7

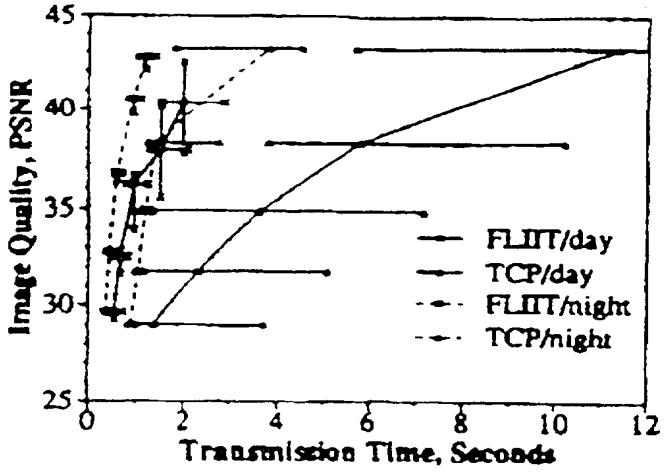


FIGURE 8A

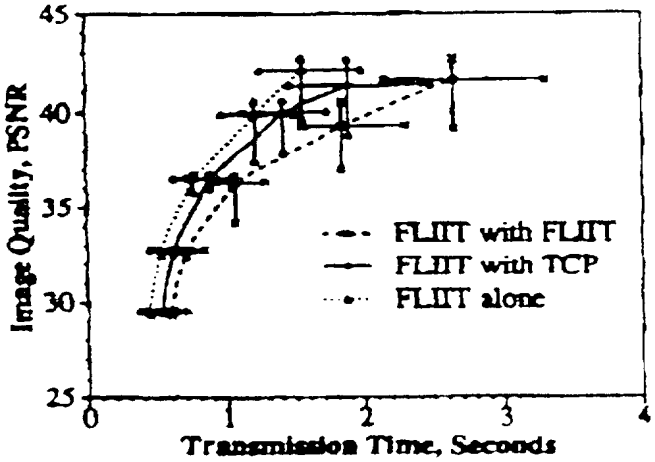
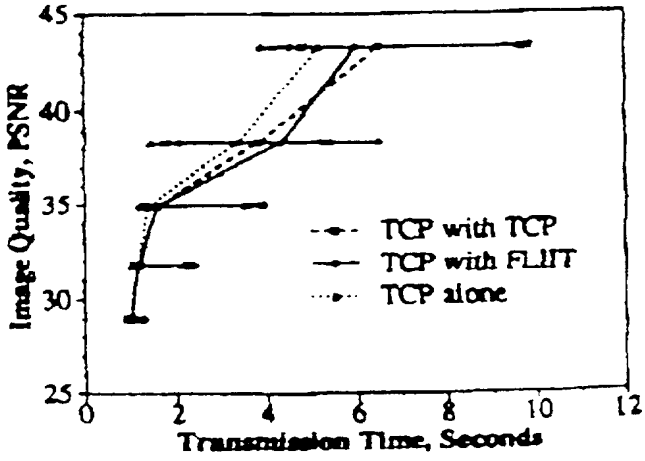


FIGURE 8B



INTERNATIONAL SEARCH REPORT

International application No.
PCT/US96/19388

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :H04N 1/41
US CL :358/426

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 358/426, 434; 348/12, 13

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,127,021 (SCHREIBER) 30 June 1992, Abstract, col. 9, line 58 - col. 10, line 8.	1-3, 16, 20, 22-23, 28, 32-34, 39, 58
Y	US 5,285,470 (SCHREIBER) 08 February 1994, Abstract, col. 11, lines 9-28.	1-3, 16, 20, 22-23, 28, 32-34, 39, 58
A	US 5,128,776 (SCORSE et al) 07 July 1992, Abstract, col. 8, lines 21-24.	1-61
A	US 5,208,682 (AHMED) 04 May 1993, Abstract, col. 3, lines 15-17.	1-61

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be part of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search

22 APRIL 1997

Date of mailing of the international search report

13 MAY 1997

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer
STEPHEN BRINICH

Telephone No. (703) 305 4390

Form PCT/ISA/210 (second sheet)(July 1992)*